

TRX433 Thermometer System

LX1BW, claude.baumann@education.lu

February 10, 2025

Start of the document: February 2, 2025

Version: 1.0

Abstract

This project focuses on the development of a temperature monitoring system for a water pipe running through an attic, aimed at preventing frost damage. The system is built around two recycled TRX433 transceiver modules (discontinued article 190045, Conrad.de, sold then at remarkable 70 bucks each!), which serve as the communication interface between the sender and receiver units. The sender module uses an NTC temperature sensor to monitor the temperature of the pipe and also measures the battery voltage through an ADC, controlled by a PIC16F88 microcontroller. The collected data is transmitted via a handshake protocol using the TRX433 transceiver on the 433 MHz frequency.

The receiver unit, equipped with a second TRX433 module, receives the transmitted data and forwards it to a second PIC16F88 microcontroller, which transmits the information to a display module via UART. The display module consists of a recycled 4-digit 7-segment display, controlled by a third PIC16F88. Additionally, the system provides visual feedback on the status of the radio connection and battery level through LEDs.

To ensure long-lasting operation, the sender is designed to consume minimal power, maximizing battery life. This low-power design approach helps maintain the system's functionality over extended periods, ensuring early detection of temperature fluctuations that could indicate a risk of freezing.

Contents

I	Introduction	2
II	Device modules	3
1	TRX433 Transceiver	3
1.1	Handshake protocol: Host→TRX433	4
1.2	Handshake protocol: TRX433→Host	5
2	4-digit 7-segment LED display	7
2.1	Common cathode display 3641AS	7
2.2	74hc595 serial-to-parallel shift register	7
2.3	Interrupt driven state machine (PIC16F88)	8
3	NTC thermistor	8
4	Battery survey	9
III	Thermometer sensor and 433MHz sender	10
5	Device schematics	10
6	PICLab Program (TX_1.0)	11
7	Assembler code (TX_1.0)	13

IV	433MHz receiver and 4-digit display	23
8	Device schematics	24
9	PICLab Program (RX_1.0) for IC1	25
10	Assembler code (RX_1.0) for IC1	25
11	PICLab Program (DISPLAY_1.0) for IC2	34
12	Assembler code (DISPLAY_1.0) for IC2	34

Preliminary note: The text of this report was partially generated by ChatGPT based on a dialogue involving keywords and preliminary explanations. However, the author and designer of the system described herein confirms that all design decisions were made independently, based on personal experience, with no external assistance other than datasheets.

Part I

Introduction

The development of a temperature monitoring system for a water pipe running through an attic is driven by both practical and ecological considerations. Several years ago, a well-insulated cold water line was installed in the attic. Although the installation has not posed any problems thus far, there has always been some uncertainty regarding the potential for frost damage during colder months. This uncertainty has led to the desire for a reliable and continuous temperature monitoring system that can detect and prevent freezing conditions before they cause any harm.

The main focus of the system described here is on cost-effectiveness and ecological sustainability. By utilizing available electronic components that were no longer in use, the project was designed with a zero-cost approach, leveraging recycled technology to minimize environmental impact. This design philosophy ensures that the system remains both affordable and environmentally friendly, aligning with the growing demand for eco-conscious technology solutions.

One of the most significant challenges of the project was ensuring reliable operation without the need for an external power supply. The sensor module, once completed, will be situated in the attic, far from any direct power sources, relying instead on a 4.5V battery pack. The system is designed for a remarkable full year autonomy on a single charge of approximately 2750mAh, a goal achieved through the use of ultra-low-power (nanoWatt) technology, specifically the PIC16F88 microcontroller. This microcontroller's Sleep mode, which consumes minimal power, is critical for maintaining long battery life. The system's wake-up process is controlled by a Watchdog Timer, which triggers the system to wake up every five minutes to initialize the TRX433 transceiver module. This module, operating at its lowest transmission power (default), sends data over the 433.9 MHz frequency channel. The entire process is designed to minimize energy consumption while ensuring timely data transmission.

The use of the TRX433 transceiver modules and the PIC16F88 microcontroller was a natural choice, as both components were readily available and have proven to be highly reliable. The TRX433 is well-regarded for its robust communication protocol, ensuring stable and efficient data transmission. Additionally, Microchip's PIC microcontrollers are known for their longevity and reliability, which is critical for applications like this that require continuous, maintenance-free operation. For instance, a complex project undertaken by the author, involving an RS03X module from BITT Technology for monitoring environmental radioactivity, has been running uninterrupted for 20 years without any maintenance.

In the development of this system, an experimental graphical code generator called PICLAB was used. Developed by the author and Dr. Laurent Kneip¹ in 2006 for the CEEO Institute at Tufts University, PICLAB is a tool designed for a selection of Microchip PIC microcontrollers. The software,

¹<https://ch.linkedin.com/in/laurent-kneip-72518658>

based on LABVIEW, generates PIC microcontroller assembler code from an intuitive icon-based flowchart. This approach allows for easy and rapid code generation while ensuring that the resulting assembler code is concise and efficient. The graphical nature of PICLAB makes it particularly useful for streamlining the development process and optimizing the design phase. Note that PICLAB is very comparable to ROBOLAB for LEGO MINDSTORMS.

Another interesting design choice was the use of a 4-digit 7-segment display for the receiver module, as opposed to more modern OLED or standard 1602 LCD displays. The decision was made with both functionality and aesthetics in mind. The display is housed in a decorative, well-placed enclosure that is designed to be easily visible. The 7-segment LEDs not only provide a nostalgic and appealing look but also offer excellent readability, even in low-light conditions, making them ideal for use in the intended setting.

This system, through its careful attention to energy efficiency, component reuse, and long-term reliability, represents a sustainable and practical solution for preventing frost damage to water pipes in attics, ensuring peace of mind for years to come.

Legal notice: The author of this project is a Luxembourgish radio amateur (LX1BW) holding a HAREC (Hobby Amateur Radio Examination Certificate) license. In Luxembourg, the frequency band 430-440 MHz is allocated (inexclusively) to amateur radio services.

Part II

Device modules

1 TRX433 Transceiver

The TRX433 Transceiver is an intelligent bidirectional device with the following characteristics:

- bidirectional communication
- 5 selectable channels: 433.3; 433.6; 433.9; 434.2; 434.5 MHz
- 3 selectable transmission power levels 0..10dBm
- maximal range: 500m
- single master and multi master operation (all devices addressable)
- internal 19200bps Manchester coding (radio signals)
- internal checksum control
- selectable 8 or 16-bytes data packages
- user-interface to host via 3 bidirectional ports (serial, synchronous with handshake)

Connections:

1. +5V (red)
2. GND (black 2x)
3. CLK (yellow)
4. ACKN (green)
5. DATA (blue)
6. $\lambda/4$ antenna (black)

1.1 Handshake protocol: Host→TRX433

The communication between the Host and the TRX433 module is managed through three lines: CLK, ACK, and DATA. To send data to the TRX433 module, the Host must control the CLK line by setting it to output (LOW). It's crucial to ensure that the CLK pin does not have any pull-up resistor. The Host then initiates the process by sending a CLK (HIGH) signal and waits for the corresponding ACK (HIGH). Next, the Host sends a CLK (LOW) signal and waits for the ACK (LOW). After this, the DATA pin should be set to output (LOW), allowing the transmission of data bytes (cf. Fig. 1-3).

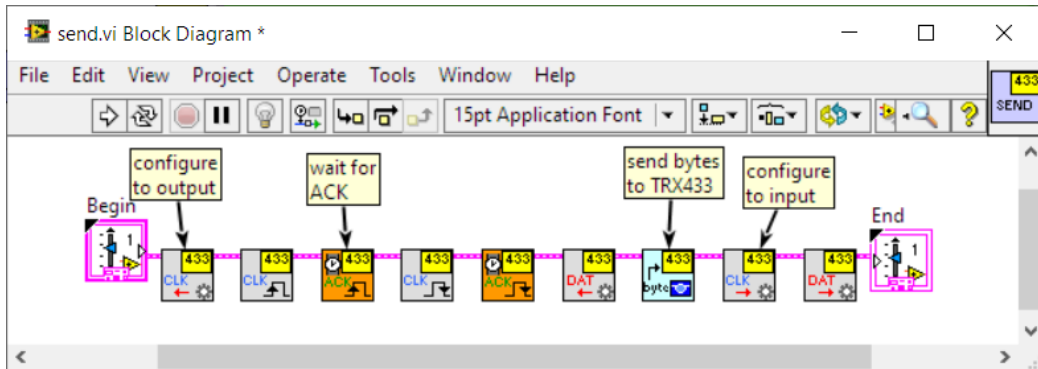


Figure 1: PICLab Flowchart: HOST→TRX433 handshake (Dataflow from the left to the right).

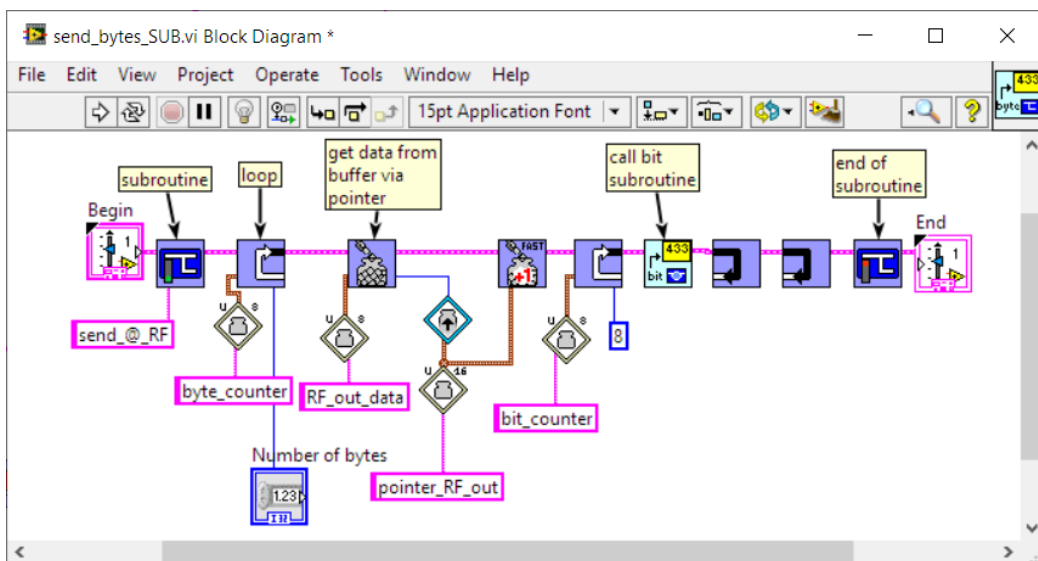


Figure 2: Send bytes from HOST→TRX433.

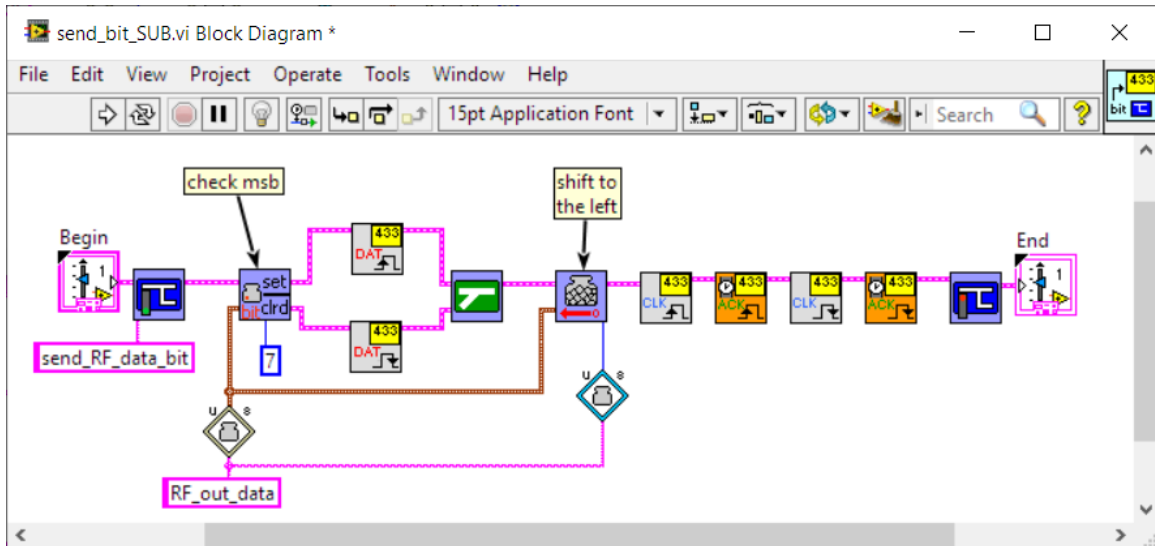


Figure 3: Set DATA for sending a single bit from HOST->TRX433.

The TRX433 module responds to a specific set of 4-byte commands from the Host, each beginning with the device address (default: 0x00). For detailed information, please consult the TRX433 user manual. 8-byte packets received from the Host that start with a non-zero address value (default 0xFF) are immediately transmitted via radio, resulting in an effective 7-byte payload. The TRX433 module automatically replaces the first byte by the default device address, which appears to have a broadcasting effect, allowing any other TRX433 module in default mode to receive the data.

In this project, only COMMAND 13 (0x00; 0x0D; 0x00; 0x00) is relevant. This command instructs the TRX433 module to transmit the most recent 8-byte RF-received packet.

Additionally, the Host is notified of an RF packet reception through the DATA (HIGH) state.

1.2 Handshake protocol: TRX433->Host

The corresponding reception handshake protocol are depicted in Fig. 4-6.

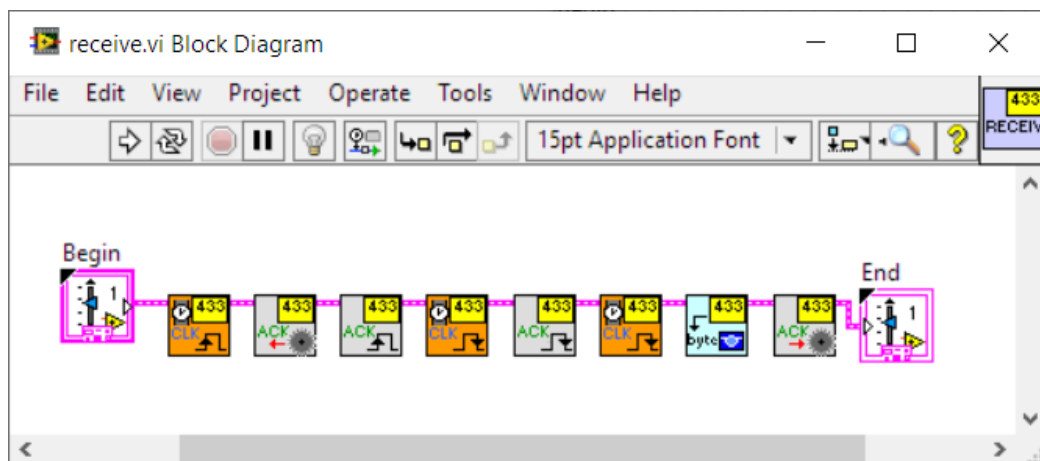


Figure 4: PICLab Flowchart: TRX433->Host handshake.

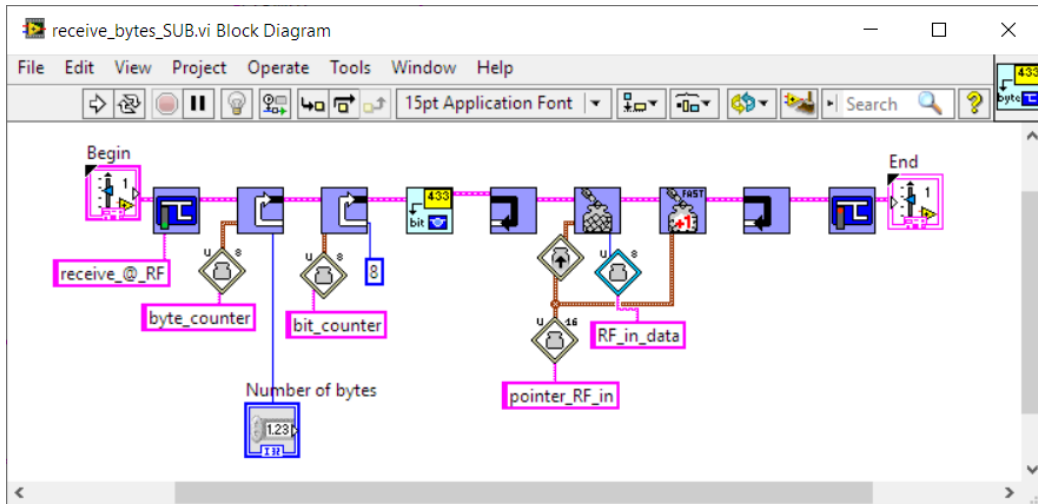


Figure 5: Get bytes from TRX433->Host.

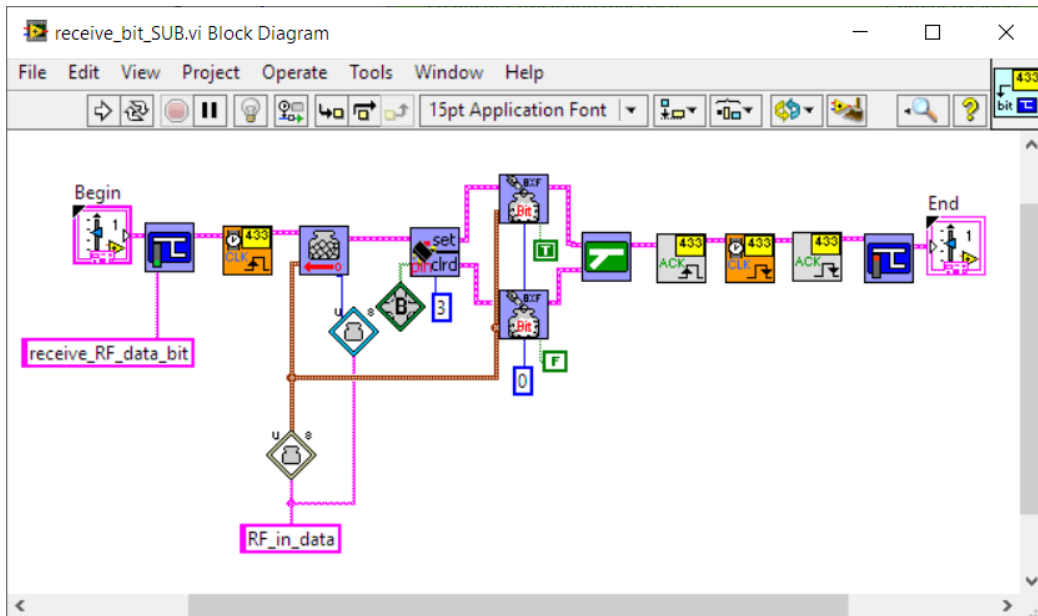


Figure 6: Single bit procedure from HOST->TRX433.

Fig. 7 shows the 3-line handshake sequence of a COMMAND 13 message followed by a packet reception.

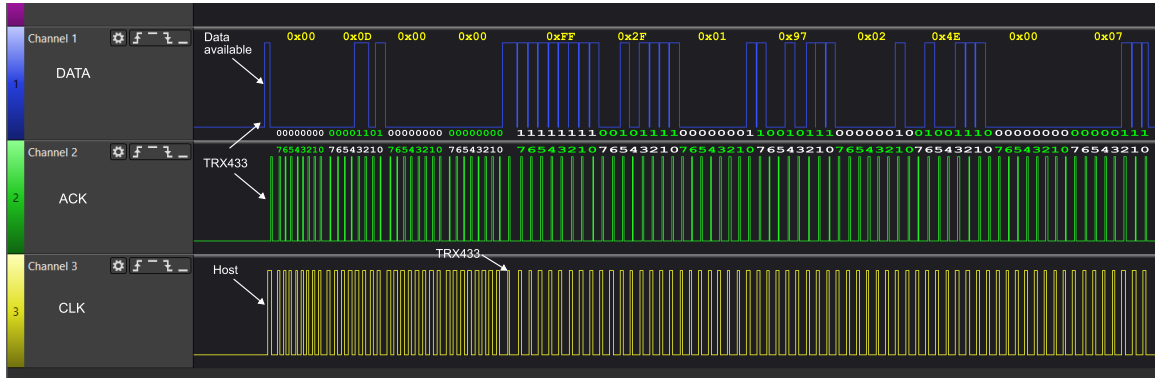


Figure 7: The TRX433 module notifies the RF reception through DATA (HIGH). The Host initiates the sending process, sends COMMAND 13, and passes to the 8-byte reception process.

2 4-digit 7-segment LED display

2.1 Common cathode display 3641AS

Fig. 8 shows an extract of the display datasheet.

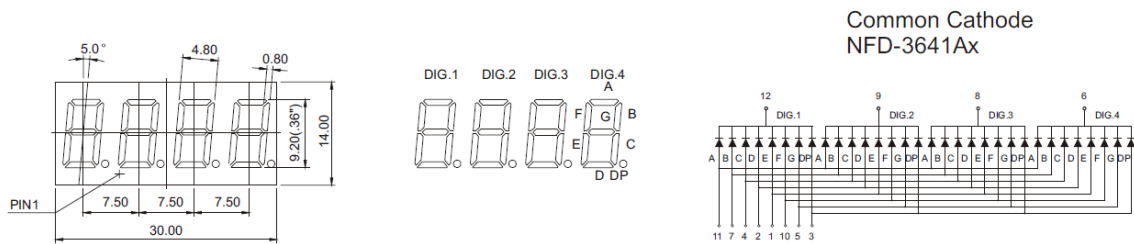


Figure 8: Common cathode 4-digit 7-segment display.

As observed in several other projects, this type of display operates by rapidly cycling through cathode sets (pins 12, 9, 8, and 6). When the switching frequency exceeds 24-30Hz, the human eye and its visual processing system can no longer perceive individual frames, causing the display to appear static.

2.2 74hc595 serial-to-parallel shift register

The display actually requires control of 8 segments per digit, including the decimal point. A common approach to manage these segments is by using a 74HC495 serial-to-parallel shift register. This integrated circuit (IC) is typically controlled via three pins, commonly referred to as A, B, and C. The 74HC495 is an 8-bit serial-in, parallel-out shift register with output latches. It allows for efficient control of multiple outputs using a minimal number of control pins. The three primary control pins are:

- SDI (Serial Data Input, pin 14): This pin receives the serial data that will be shifted into the register.
- SRCLK (Shift Register Clock Input, pin 11): Also known as the shift clock, this pin controls the timing of data shifting into the register.
- RCLK (Storage Register Clock Input, pin 12): Also known as the latch clock, this pin controls when the shifted data is transferred to the output latches, making it available at the output pins.

By manipulating these pins, you can efficiently control the 8 segments of the display, including the decimal point, with just three control lines:

1. disable data latch to output: RCLK (LOW)
2. repeat 8 times:
 - SRCLK (LOW)
 - set or clear SDI as desired
 - short delay
 - SRCLK (HIGH)
3. latch data to parallel output: RCLK (HIGH)

2.3 Interrupt driven state machine (PIC16F88)

Fig. 9 illustrates a timer interrupt-controlled state machine designed to continuously transmit data to the display. To ensure a stable overall program on the controlling microcontroller, the microcontroller remains in the Interrupt Service Routine (ISR) for a very brief period, transitioning to a different state with each interrupt. This design results in a display digit cycle corresponding to two timer overflows for RCLK, and sixteen timer overflows per data bit (SRCLK and SDI), totaling 18 timer overflows. The timer is configured to trigger an overflow interrupt every 0.25 milliseconds, leading to a display frequency of $1/(4 \cdot 18 \cdot 0.25\text{ms}) = 55.55\text{Hz}$. This configuration ensures flicker-free operation of the display.

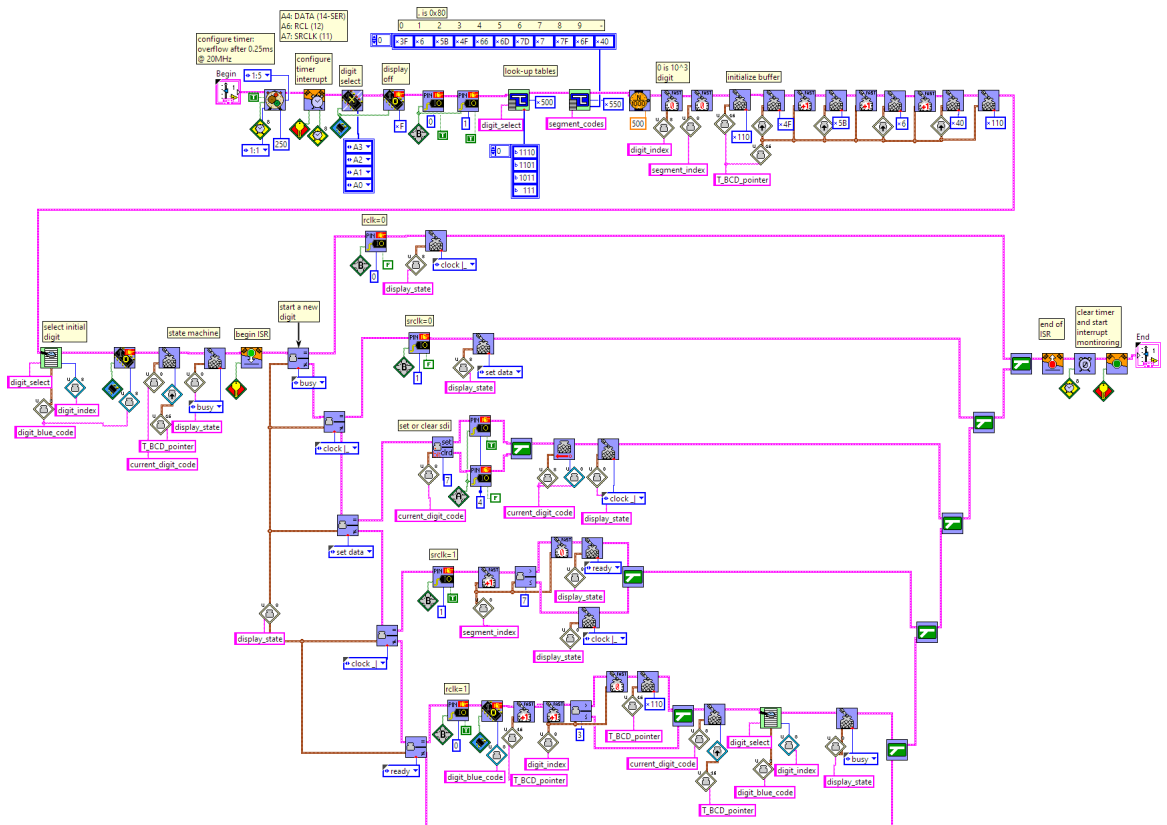


Figure 9: Timer interrupt controlled state machine.

3 NTC thermistor

We use a NTC thermistor 10k at 25°C. Fig. 10 shows the calibration curve for 10-bit ADC readings with 10k pull-up resistor added. Since the 8-bit PIC16F88 can only handle 16-bit integer variables

with PICLAB, it is essential to implement a reliable conversion from ADC values to temperature. During the data manipulation for calculation, it is crucial to ensure that the 16-bit boundaries are never exceeded.

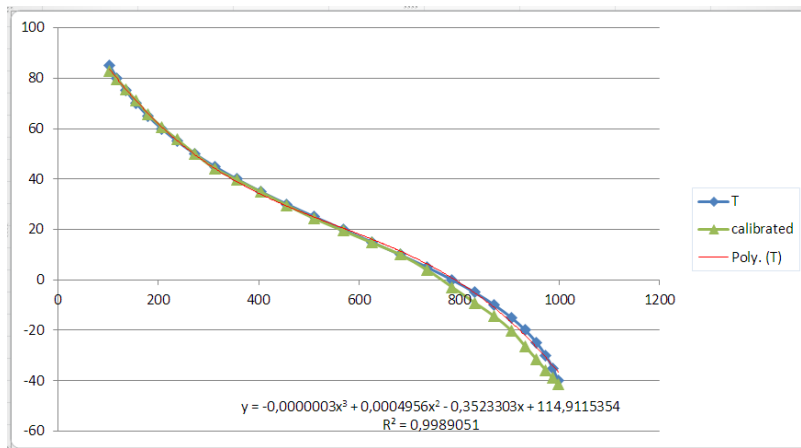


Figure 10: Calibration curve of the NTC.

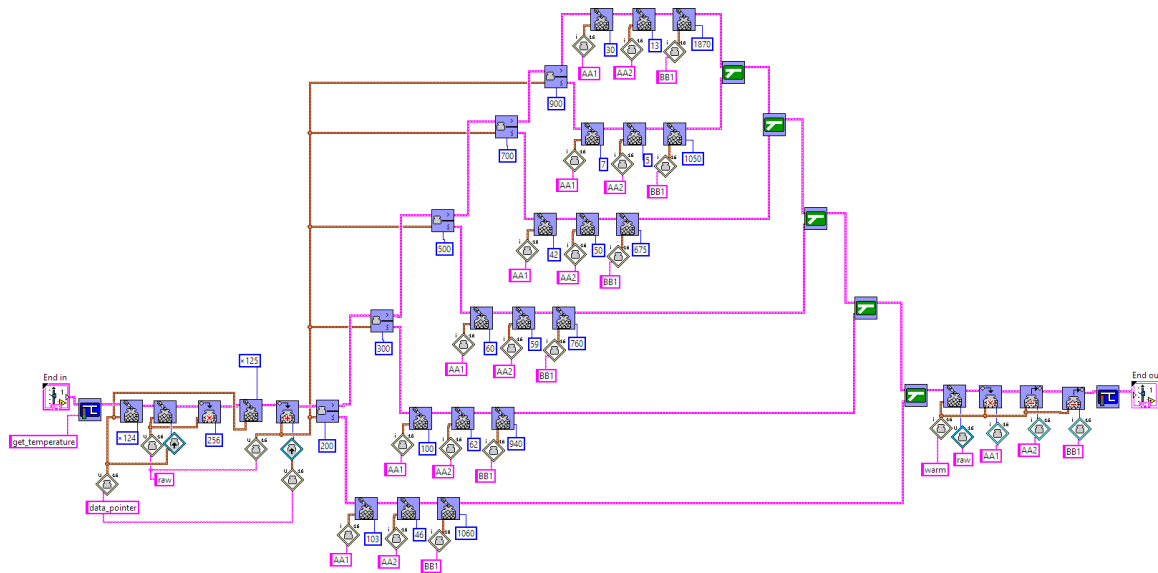


Figure 11: The segmented linear conversion of raw ADC data to °C.

4 Battery survey

The transmitter of this wireless thermometer, which is placed in the attic and has no direct connection to the power grid, relies on battery autonomy. To ensure the device operates for as long as possible without manual intervention, the battery level of three standard AAA alkaline batteries connected in series is continuously monitored. These batteries provide an approximate voltage of 4.5V when fully charged.

The Microcontroller Unit (MCU), which in this case is a PIC16F88, is responsible for measuring the battery voltage. However, there's a challenge: the battery voltage also serves as the reference voltage for the Analog-to-Digital Converter (ADC). A simple series resistor bridge would not provide any useful measurements, as the ADC-converted voltage across the resistors would remain constant, regardless of the battery's charge level. Note that this property is essential for accurate temperature measurement of the NTC thermistor in series with a 10k resistor, ensuring that the temperature

readings remain independent of the battery level.

To overcome this, one resistor is replaced with three 1N4148 diodes connected in series. The diodes create a voltage drop of approximately 0.6V per diode, totaling about 1.8V for the three diodes. This voltage drop remains nearly constant, regardless of the applied battery voltage. However, as the battery drains and the reference voltage changes, the corresponding ADC values shift in accordance with the changes in the battery voltage (inversely proportionally), enabling accurate monitoring of the battery's charge level. This allows for accurate mapping of the voltage to 10-bit ADC values, as shown in the table below (cf. Table 1).

10-bit ADC	Battery Voltage
495 (0x1ef)	3.0V
438 (0x1b6)	3.5V
392 (0x188)	4.0V
355 (0x163)	4.5V
324 (0x144)	5.0V

Table 1: ADC Values and Corresponding Battery Voltage

Part III

Thermometer sensor and 433MHz sender

5 Device schematics

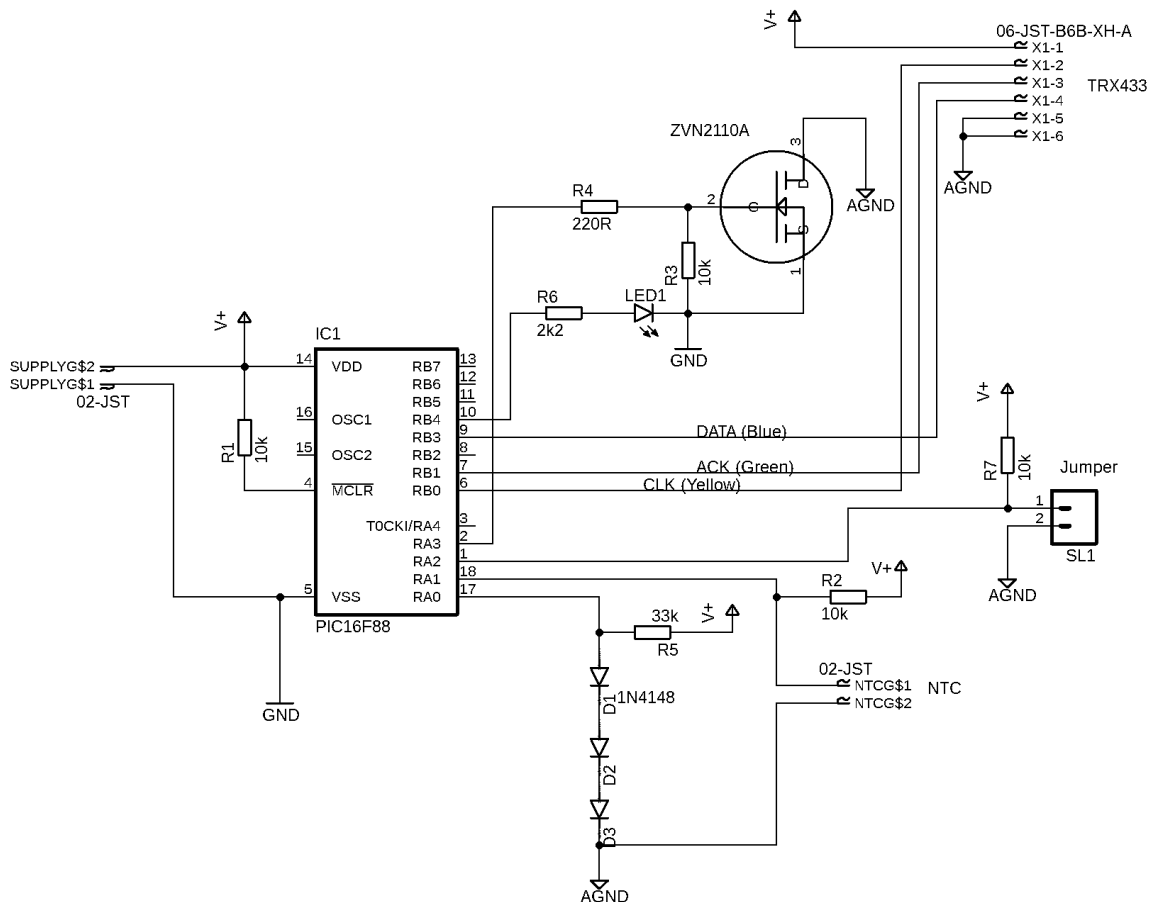


Figure 12: Sender device schematics.

The communication to the TRX433 module is controlled by the PIC16F88 microcontroller. When the jumper SL1 is set, the microcontroller triggers communication approximately every 8 seconds, while without the jumper, communication occurs every 5 minutes. The PIC16F88 utilizes its watchdog timer to power the TRX433 module on via a low-side MOSFET switch. The temperature and battery voltage are measured using the ADC converter and transmitted to the receiver. The MOSFET is controlled by the RA3 pin of the PIC16F88 through a protection resistor.

The TRX433 module's CLK, ACK, and DATA lines are controlled by pins RB0, RB1, and RB3 of the PIC16F88. Additionally, pin RB4 is responsible for a status LED that briefly lights up when the TRX433 module is transmitting. Battery voltage is measured according to Section 4, and the MCLR pin of the PIC16F88 is equipped with a 10k pull-up resistor. The PIC16F88 is clocked using its internal 8 MHz oscillator, and the watchdog timer (WDT) is enabled to ensure system stability and to wake the PIC from sleep mode regularly.

Setting the jumper SL1 is primarily useful during device testing. Under normal operation without the jumper set, the current consumption is 15mA during the active phase when the PIC, its ADC module, LED, and the TRX433 module are operating. This phase lasts for about 5 seconds. During the remaining 5-minute period, all peripheral modules are switched off, and the PIC is woken up every 2.3 seconds from sleep by the watchdog timer for a few microseconds before going back to sleep. Current consumption drops to $4\mu A$ during this time. The average current during this 5-minute cycle can be calculated as follows:

$$\text{Average Current} = \frac{5 \cdot 15 + 295 \cdot 4 \cdot 10^{-3}}{300} = 0.25mA \quad (1)$$

With a 2750mAh battery pack, the battery life span can be estimated as follows:

$$\text{Battery Life Span} = \frac{2750}{0.25} = 11000h > 1 \text{ year} \quad (2)$$

6 PICLab Program (TX_1.0)

The PIC16F88 microcontroller (MCU) is configured to operate with three analog-to-digital converter (ADC) channels and communicate with the TRX433 module. The main program functions as a state machine with the following states:

1. Initialization:

- The MCU exits Sleep mode.
- Clear variables and pointers
- The TRX433 module is powered on, with a 4-second power-up time.
- Transition to the next state.

2. ADC:

- Analog channels are read, and data is stored in a buffer.
- The jumper is checked:
- If set, a short power-down time is initiated.
- If not set, a long power-down time is initiated.
- Transition to the next state.

3. RF_Send:

- Data is transmitted to the TRX433 module, which automatically forwards it via radio using Manchester encoding.
- The TRX433 module is powered off.
- Transition to the next state.

4. Lazy:

- The MCU enters Sleep mode for 2.3 seconds.

- Wake-up phases are counted based on the jumper setting;
- After the desired number of wake-up phases, the MCU returns to the Initialization state.
- The cycle repeats.

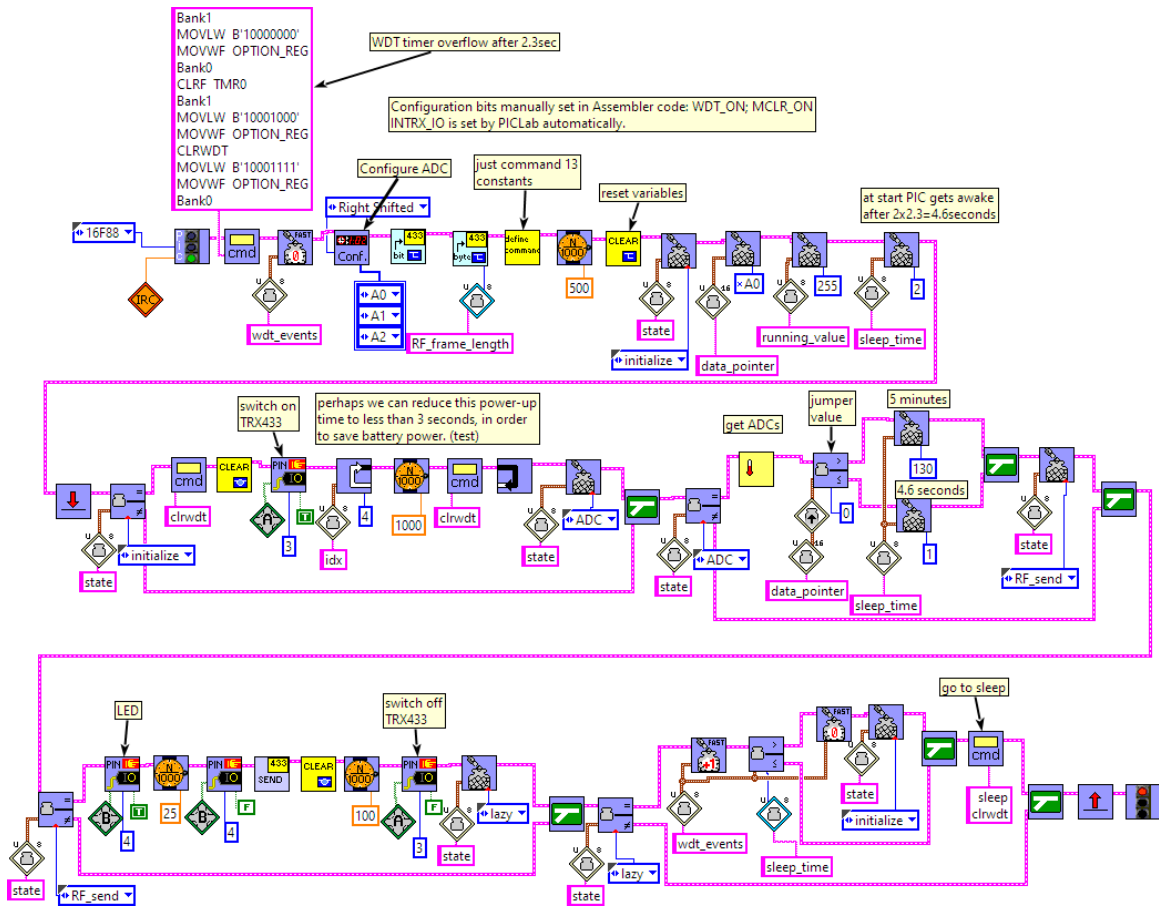


Figure 13: PICLAB main program flowchart (LV filename: *test_433-TX5.vi*).

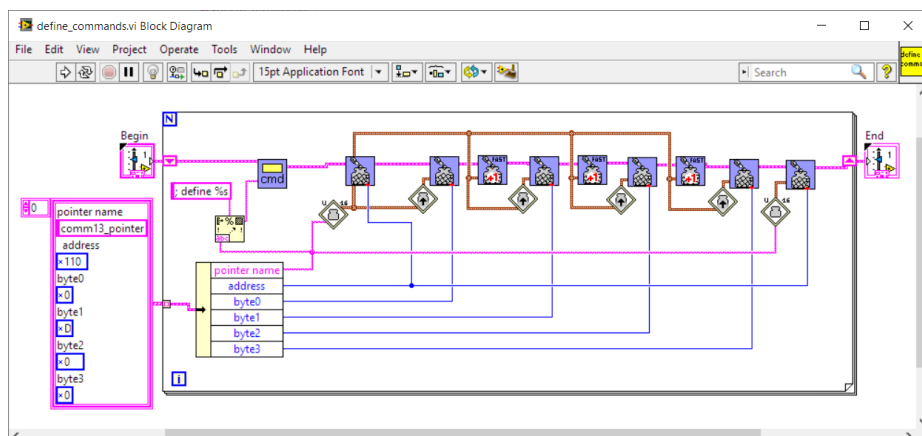


Figure 14: This sub.vi represents a makro defining several TRX433 commands (actually only COMMAND 13 is used).

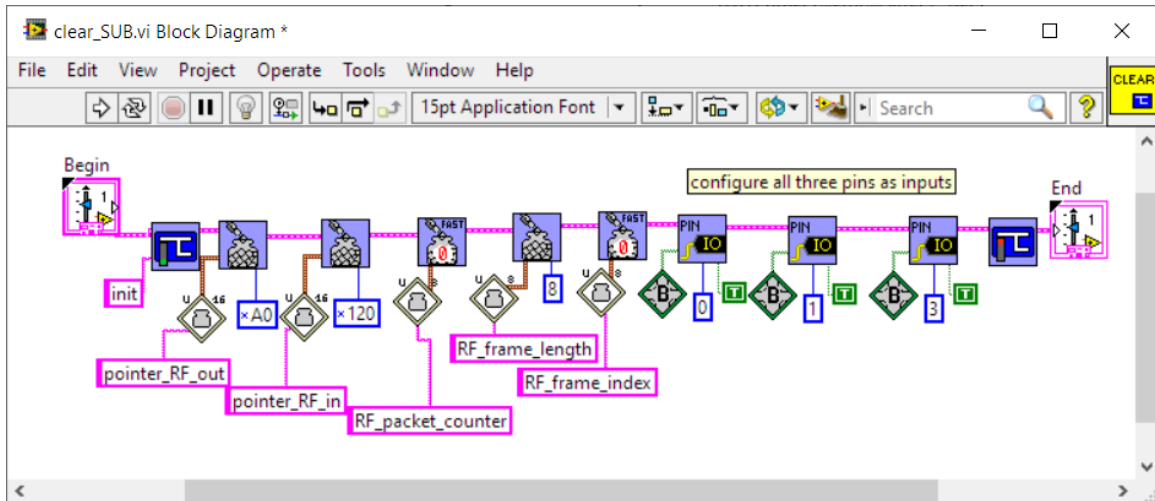


Figure 15: This sub.vi represents a PIC subroutine that is regularly called in order to reset variables. The communication pins to the TRX433 module are configured as inputs.

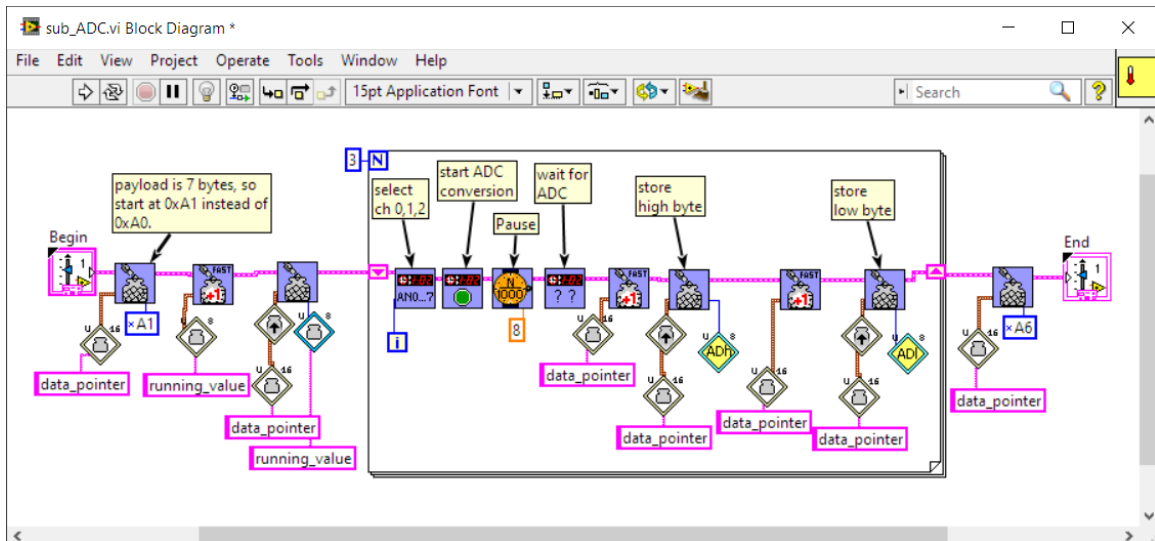


Figure 16: This sub.vi represents a makro defining the ADC and buffer storage process. Note that the first payload value is a running counter helping to identify packets.

7 Assembler code (TX_1.0)

```

1  ;-----
2  ;                               Assembler code for test_433_TX5.vi created with PICLab
3  ;-----
4
5  LIST    p=16F88
6
7  #include "P16F88.INC" ; Include header file
8
9  __CONFIG _CONFIG1, _BODEN_OFF&_CP_OFF&_PWRT_ON&_WDT_ON&_LVP_OFF&_MCLR_ON&_INTRC_IO
10
11 ;*****Variable definitions*****
12
13 TEMPPORT8 EQU 0X20
14 TEMPX8 EQU 0X21
15 TEMPY8 EQU 0X22
16 RESULT8 EQU 0X23
17 TEMPX16_H EQU 0X24
18 TEMPY16_H EQU 0X25
19 TEMPY16 EQU 0X26
20 TEMPY16_H EQU 0X27
21 RESULT16 EQU 0X28
22 RESULT16_H EQU 0X29
23 IDX16 EQU 0X2A

```

```

24 | IDX16_H EQU 0X2B
25 | TEMPPY EQU 0X2C
26 | OP_SIGN8 EQU 0X2D
27 | ADHBYTE EQU 0X2E
28 | ADLBYTE EQU 0X2F
29 | AD_RESULT EQU 0X30
30 | AD_RESULT_H EQU 0X31
31 | ISR_TEMPPORT8 EQU 0X32
32 | ISR_TEMPX8 EQU 0X33
33 | ISR_TEMPY8 EQU 0X34
34 | ISR_RESULTS EQU 0X35
35 | ISR_TEMPX16 EQU 0X36
36 | ISR_TEMPX16_H EQU 0X37
37 | ISR_TEMPY16 EQU 0X38
38 | ISR_TEMPY16_H EQU 0X39
39 | ISR_RESULT16 EQU 0X3A
40 | ISR_RESULT16_H EQU 0X3B
41 | ISR_IDX16 EQU 0X3C
42 | ISR_IDX16_H EQU 0X3D
43 | ISR_TEMPPY EQU 0X3E
44 | ISR_OP_SIGN8 EQU 0X3F
45 | ISR_STATUS EQU 0X40
46 | ISR_W EQU 0X41
47 | ISR_FSR EQU 0X42
48 | SLEEP_TIME EQU 0X43
49 | RF_FRAME_LENGTH EQU 0X44
50 | DATA_POINTER EQU 0X45
51 | DATA_POINTER_H EQU 0X46
52 | RUNNING_VALUE EQU 0X47
53 | WDT_EVENTS EQU 0X48
54 | STATE EQU 0X49
55 | IDX EQU 0X4A
56 | RF_OUT_DATA EQU 0X4B
57 | BIT_COUNTER EQU 0X4C
58 | POINTER_RF_OUT EQU 0X4D
59 | POINTER_RF_OUT_H EQU 0X4E
60 | BYTE_COUNTER EQU 0X4F
61 | COMM13_POINTER EQU 0X50
62 | COMM13_POINTER_H EQU 0X51
63 | COMM6_POINTER EQU 0X52
64 | COMM6_POINTER_H EQU 0X53
65 | COMM7_POINTER EQU 0X54
66 | COMM7_POINTER_H EQU 0X55
67 | RF_FRAME_INDEX EQU 0X56
68 | RF_PACKET_COUNTER EQU 0X57
69 | POINTER_RF_IN EQU 0X58
70 | POINTER_RF_IN_H EQU 0X59
71 |
72 | ;*****Makro definitions and definitions of used operations*****
73 |
74 | BANK0 MACRO
75 |     BCF STATUS,RP0
76 |     BCF STATUS,RP1
77 |     ENDM
78 | BANK1 MACRO
79 |     BSF STATUS,RP0
80 |     BCF STATUS,RP1
81 |     ENDM
82 | BANK2 MACRO
83 |     BCF STATUS,RP0
84 |     BSF STATUS,RP1
85 |     ENDM
86 | BANK3 MACRO
87 |     BSF STATUS,RP0
88 |     BSF STATUS,RP1
89 |     ENDM
90 |
91 |             GOTO START
92 |             ORG 0X4
93 |
94 | SHLV8
95 |     BCF STATUS,C
96 |     RLF TEMPY8,F
97 |     RETURN
98 |
99 | EQUUS8
100 |     MOVF     TEMPX8,W
101 |     SUBWF   TEMPY8
102 |     BTFSS   STATUS,Z ;SKIP IF ZERO ->IDENTICAL
103 |     RETLW  0
104 |     RETLW  1
105 |
106 | GRUS8
107 |     MOVF     TEMPX8,W
108 |     SUBWF   TEMPY8
109 |     BTFSS   STATUS,C
110 |     RETLW  0
111 |     RETLW  1
112 |
113 | ;*****BEGIN OF MAIN PROGRAM*****
114 |
115 | START
116 |
117 | ; INITIALIZE PORT A
118 | BANK0
119 |     CLRF PORTA
120 | BANK1
121 |     CLRF ANSEL
122 |     MOVLW 0X07
123 |     MOVWF CMCON
124 |     MOVLW 0X0
125 |     MOVWF TRISA
126 |
127 | ; INITIALIZE PORT B
128 | BANK0
129 |     CLRF PORTB
130 | BANK1
131 |     MOVLW 0X0
132 |     MOVWF TRISB
133 |
134 | ; CONFIGURE INTERNAL OSCILLATOR
135 |     MOVLW 0XFC
136 |     MOVWF OSCCON
137 |
138 |     MOVLW B'10000000'
139 |     MOVWF OPTION_REG
140 | BANK0
141 |     CLRF TMRO
142 | BANK1

```

```

143 MOVW B'10001000'
144 MOVWF OPTION_REG
145 CLRWDI
146 MOVW B'10001111'
147 MOVWF OPTION_REG
148 BANK0
149
150 CLRWF WDT_EVENTS
151
152 ;CONFIGURE AD-CONVERTER
153 ;CONFIGURE PORT INPUTS
154 BANK1
155 MOVF TRISA,W
156 IORLW 0X7
157 MOVWF TRISA
158 MOVW 0X7
159 MOVWF ANSEL
160 MOVW 0XCO
161 MOVWF ADCON1
162 MOVW 0X41
163 BANK0
164 MOVWF ADCON0
165
166 ; DEFINE COMM13_POINTER
167
168 ;SET VARIABLE U16 WITH CONSTANT
169 MOVW 0X10
170 MOVWF COMM13_POINTER
171 MOVW 0X1
172 MOVWF COMM13_POINTER_H
173
174 ;SET INDIRECT VARIABLE WITH CONSTANT
175 ;SET U16 POINTER
176 BCF STATUS,IRP
177 BTFSC COMM13_POINTER_H,0
178 BSF STATUS,IRP
179 BANK0
180 MOVF COMM13_POINTER,W
181 MOVWF FSR
182 MOVW 0X0
183 MOVWF INDF
184
185 INCF COMM13_POINTER,F
186 BTFSC STATUS,Z
187 INCF COMM13_POINTER_H,F
188
189 ;SET INDIRECT VARIABLE WITH CONSTANT
190 ;SET U16 POINTER
191 BCF STATUS,IRP
192 BANK0
193 BTFSC COMM13_POINTER_H,0
194 BSF STATUS,IRP
195 BANK0
196 MOVF COMM13_POINTER,W
197 MOVWF FSR
198 MOVW 0XD
199 MOVWF INDF
200
201 INCF COMM13_POINTER,F
202 BTFSC STATUS,Z
203 INCF COMM13_POINTER_H,F
204
205 ;SET INDIRECT VARIABLE WITH CONSTANT
206 ;SET U16 POINTER
207 BCF STATUS,IRP
208 BANK0
209 BTFSC COMM13_POINTER_H,0
210 BSF STATUS,IRP
211 BANK0
212 MOVF COMM13_POINTER,W
213 MOVWF FSR
214 MOVW 0X0
215 MOVWF INDF
216
217 INCF COMM13_POINTER,F
218 BTFSC STATUS,Z
219 INCF COMM13_POINTER_H,F
220
221 ;SET INDIRECT VARIABLE WITH CONSTANT
222 ;SET U16 POINTER
223 BCF STATUS,IRP
224 BANK0
225 BTFSC COMM13_POINTER_H,0
226 BSF STATUS,IRP
227 BANK0
228 MOVF COMM13_POINTER,W
229 MOVWF FSR
230 MOVW 0X0
231 MOVWF INDF
232
233 ;SET VARIABLE U16 WITH CONSTANT
234 MOVW 0X10
235 MOVWF COMM13_POINTER
236 MOVW 0X1
237 MOVWF COMM13_POINTER_H
238
239 ; DEFINE COMM6_POINTER
240
241 ;SET VARIABLE U16 WITH CONSTANT
242 MOVW 0X14
243 MOVWF COMM6_POINTER
244 MOVW 0X1
245 MOVWF COMM6_POINTER_H
246
247 ;SET INDIRECT VARIABLE WITH CONSTANT
248 ;SET U16 POINTER
249 BCF STATUS,IRP
250 BTFSC COMM6_POINTER_H,0
251 BSF STATUS,IRP
252 BANK0
253 MOVF COMM6_POINTER,W
254 MOVWF FSR
255 MOVW 0X0
256 MOVWF INDF
257
258 INCF COMM6_POINTER,F
259 BTFSC STATUS,Z
260 INCF COMM6_POINTER_H,F
261

```

```

262 ;SET INDIRECT VARIABLE WITH CONSTANT
263 ;SET U16 POINTER
264 BCF STATUS,IRP
265 BANKO
266 BTFSK COMM6_POINTER_H,0
267 BSF STATUS,IRP
268 BANKO
269 MOVF COMM6_POINTER,W
270 MOVWF FSR
271 MOVLW 0X6
272 MOVWF INDF
273
274 INCF COMM6_POINTER,F
275 BTFSK STATUS,Z
276 INCF COMM6_POINTER_H,F
277
278 ;SET INDIRECT VARIABLE WITH CONSTANT
279 ;SET U16 POINTER
280 BCF STATUS,IRP
281 BANKO
282 BTFSK COMM6_POINTER_H,0
283 BSF STATUS,IRP
284 BANKO
285 MOVF COMM6_POINTER,W
286 MOVWF FSR
287 MOVLW 0X6F
288 MOVWF INDF
289
290 INCF COMM6_POINTER,F
291 BTFSK STATUS,Z
292 INCF COMM6_POINTER_H,F
293
294 ;SET INDIRECT VARIABLE WITH CONSTANT
295 ;SET U16 POINTER
296 BCF STATUS,IRP
297 BANKO
298 BTFSK COMM6_POINTER_H,0
299 BSF STATUS,IRP
300 BANKO
301 MOVF COMM6_POINTER,W
302 MOVWF FSR
303 MOVLW 0X0
304 MOVWF INDF
305
306 ;SET VARIABLE U16 WITH CONSTANT
307 MOVLW 0X14
308 MOVWF COMM6_POINTER
309 MOVLW 0X1
310 MOVWF COMM6_POINTER_H
311
312 ; DEFINE COMM7_POINTER
313
314 ;SET VARIABLE U16 WITH CONSTANT
315 MOVLW 0X18
316 MOVWF COMM7_POINTER
317 MOVLW 0X1
318 MOVWF COMM7_POINTER_H
319
320 ;SET INDIRECT VARIABLE WITH CONSTANT
321 ;SET U16 POINTER
322 BCF STATUS,IRP
323 BTFSK COMM7_POINTER_H,0
324 BSF STATUS,IRP
325 BANKO
326 MOVF COMM7_POINTER,W
327 MOVWF FSR
328 MOVLW 0X0
329 MOVWF INDF
330
331 INCF COMM7_POINTER,F
332 BTFSK STATUS,Z
333 INCF COMM7_POINTER_H,F
334
335 ;SET INDIRECT VARIABLE WITH CONSTANT
336 ;SET U16 POINTER
337 BCF STATUS,IRP
338 BANKO
339 BTFSK COMM7_POINTER_H,0
340 BSF STATUS,IRP
341 BANKO
342 MOVF COMM7_POINTER,W
343 MOVWF FSR
344 MOVLW 0X7
345 MOVWF INDF
346
347 INCF COMM7_POINTER,F
348 BTFSK STATUS,Z
349 INCF COMM7_POINTER_H,F
350
351 ;SET INDIRECT VARIABLE WITH CONSTANT
352 ;SET U16 POINTER
353 BCF STATUS,IRP
354 BANKO
355 BTFSK COMM7_POINTER_H,0
356 BSF STATUS,IRP
357 BANKO
358 MOVF COMM7_POINTER,W
359 MOVWF FSR
360 MOVLW 0X3
361 MOVWF INDF
362
363 INCF COMM7_POINTER,F
364 BTFSK STATUS,Z
365 INCF COMM7_POINTER_H,F
366
367 ;SET INDIRECT VARIABLE WITH CONSTANT
368 ;SET U16 POINTER
369 BCF STATUS,IRP
370 BANKO
371 BTFSK COMM7_POINTER_H,0
372 BSF STATUS,IRP
373 BANKO
374 MOVF COMM7_POINTER,W
375 MOVWF FSR
376 MOVLW 0X0
377 MOVWF INDF
378
379 ;SET VARIABLE U16 WITH CONSTANT
380 MOVLW 0X18

```



```

381 MOVWF COMM7_POINTER
382 MOVLW 0X1
383 MOVWF COMM7_POINTER_H
384
385 ;WAIT
386 MOVLW 0X06
387 MOVWF TEMPPY
388 LABEL_1013
389 MOVLW 0X7
390 MOVWF IDX16
391 LABEL_1014
392 MOVLW 0XCD
393 MOVWF IDX16_H
394 LABEL_1015
395 DECFSZ IDX16_H,F
396 GOTO LABEL_1015
397 DECFSZ IDX16,F
398 GOTO LABEL_1014
399 DECFSZ TEMPPY,F
400 GOTO LABEL_1013
401 NOP
402 NOP
403 NOP
404 NOP
405 NOP
406 NOP
407 NOP
408
409 ;SET VARIABLE U8 WITH CONSTANT
410 MOVLW 0X6
411 BANK0
412 MOVWF STATE
413
414 ;SET VARIABLE U16 WITH CONSTANT
415 MOVLW 0XA0
416 MOVWF DATA_POINTER
417 MOVLW 0X0
418 MOVWF DATA_POINTER_H
419
420 ;SET VARIABLE U8 WITH CONSTANT
421 MOVLW 0XF
422 MOVWF RUNNING_VALUE
423
424 ;SET VARIABLE U8 WITH CONSTANT
425 MOVLW 0X2
426 MOVWF SLEEP_TIME
427
428 LABEL_0
429
430 ;BEGIN OF IF-STRUCTURE
431 ;COMPARE-OPERATION
432 ;SET VARIABLE U8 WITH VARIABLE U8
433 BANK0
434 MOVF STATE,W
435 MOVWF TEMPX8
436 ;SET VARIABLE U8 WITH CONSTANT
437 MOVLW 0X6
438 MOVWF TEMPY8
439 CALL EQU8U8
440 BANK0
441 MOVWF RESULT8
442 BTFSZ RESULT8,0
443 GOTO LABEL_1016
444
445 GOTO LABEL_1022
446 LABEL_1016
447
448 CLRWD
449
450 CALL LABEL_INIT
451
452 ;SET SINGLE OUPUT PIN
453 BANK0
454 BSF PORTA,3
455
456 ;BEGIN OF FOR-LOOP
457 ;SET VARIABLE U8 WITH CONSTANT
458 MOVLW 0X4
459 MOVWF IDX
460 INCF IDX,F
461 LABEL_1017
462 BANK0
463 DECF IDX,F
464 BTFSZ STATUS,Z
465 GOTO LABEL_1018
466
467 ;WAIT
468 BANK0
469 MOVLW 0XD0
470 MOVWF TEMPPY
471 LABEL_1019
472 MOVLW 0X13
473 MOVWF IDX16
474 LABEL_1020
475 MOVLW 0XA7
476 MOVWF IDX16_H
477 LABEL_1021
478 DECFSZ IDX16_H,F
479 GOTO LABEL_1021
480 DECFSZ IDX16,F
481 GOTO LABEL_1020
482 DECFSZ TEMPPY,F
483 GOTO LABEL_1019
484 NOP
485 NOP
486 NOP
487 NOP
488 NOP
489 CLRWD
490
491 GOTO LABEL_1017
492 LABEL_1018
493 ;END OF FOR/WHILE-LOOP
494
495 ;SET VARIABLE U8 WITH CONSTANT
496 MOVLW 0X3
497 BANK0
498 MOVWF STATE
499

```

```

500 LABEL_1022
501 ;END OF IF-STRUCTURE
502
503 ;BEGIN OF IF-STRUCTURE
504 ;COMPARE-OPERATION
505 ;SET VARIABLE U8 WITH VARIABLE U8
506 BANK0
507 MOVF STATE,W
508 MOVWF TEMPX8
509 ;SET VARIABLE U8 WITH CONSTANT
510 MOVLW 0X3
511 MOVWF TEMPY8
512 CALL EQU8U8
513 BANK0
514 MOVWF RESULTS8
515 BTFSC RESULTS8,0
516 GOTO LABEL_1023
517
518 GOTO LABEL_1035
519 LABEL_1023
520
521 ;SET VARIABLE U16 WITH CONSTANT
522 MOVLW 0XA1
523 BANK0
524 MOVWF DATA_POINTER
525 MOVLW 0X0
526 MOVWF DATA_POINTER_H
527
528 INCF RUNNING_VALUE,F
529
530 ;SET INDIRECT VARIABLE WITH VARIABLE U8
531 ;SET U16 POINTER
532 BCF STATUS,IRP
533 BTFSC DATA_POINTER_H,0
534 BSF STATUS,IRP
535 BANK0
536 MOVF DATA_POINTER,W
537 MOVWF FSR
538 MOVF RUNNING_VALUE,W
539 MOVWF INDF
540
541 ;SELECT AD-CHANNEL
542 BCF ADCON0,3
543 BCF ADCON0,4
544 BCF ADCON0,5
545 CALL LABEL_AQUISITION
546
547 ;START AD-CONVERSION
548 BANK0
549 BSF ADCON0,2
550
551 ;WAIT
552 MOVLW 0X1D
553 MOVWF TEMPYY
554 LABEL_1024
555 MOVLW 0XAC
556 MOVWF IDX16
557 LABEL_1025
558 DECFSZ IDX16,F
559 GOTO LABEL_1025
560 DECFSZ TEMPYY,F
561 GOTO LABEL_1024
562 NOP
563
564 ;FINISH AD-CONVERSION
565 BANK0
566 LABEL_1026
567 BTFSC ADCON0,2
568 GOTO LABEL_1026
569 ;SET VARIABLE U8 WITH SFR
570 BANK0
571 MOVF ADRESH,W
572 MOVWF ADHBYTE
573 ;SET VARIABLE U8 WITH SFR
574 BANK1
575 MOVF ADRESL,W
576 BANK0
577 MOVWF ADLBYTE
578 ;SET VARIABLE U16 WITH VARIABLE U8
579 MOVF ADLBYTE,W
580 MOVWF AD_RESULT
581 CLRf AD_RESULT_H
582 ;SET 16-BIT REGISTER HIGH BYTE THROUGH 8-BIT REGISTER
583 MOVF ADHBYTE,W
584 MOVWF AD_RESULT_H
585
586 INCF DATA_POINTER,F
587 BTFSC STATUS,Z
588 INCF DATA_POINTER_H,F
589
590 ;SET INDIRECT VARIABLE WITH VARIABLE U8
591 ;SET U16 POINTER
592 BCF STATUS,IRP
593 BANK0
594 BTFSC DATA_POINTER_H,0
595 BSF STATUS,IRP
596 BANK0
597 MOVF DATA_POINTER,W
598 MOVWF FSR
599 MOVF ADHBYTE,W
600 MOVWF INDF
601
602 INCF DATA_POINTER,F
603 BTFSC STATUS,Z
604 INCF DATA_POINTER_H,F
605
606 ;SET INDIRECT VARIABLE WITH VARIABLE U8
607 ;SET U16 POINTER
608 BCF STATUS,IRP
609 BANK0
610 BTFSC DATA_POINTER_H,0
611 BSF STATUS,IRP
612 BANK0
613 MOVF DATA_POINTER,W
614 MOVWF FSR
615 MOVF ADLBYTE,W
616 MOVWF INDF
617
618

```

```

619 ;SELECT AD-CHANNEL
620 BSF ADCON0,3
621 BCF ADCON0,4
622 BCF ADCON0,5
623 CALL LABEL_AQUISITION
624
625 ;START AD-CONVERSION
626 BANK0
627 BSF ADCON0,2
628
629 ;WAIT
630 MOVLW 0X1D
631 MOVWF TEMPYY
632 LABEL_1027
633 MOVLW 0XAC
634 MOVWF IDX16
635 LABEL_1028
636 DECFSZ IDX16,F
637 GOTO LABEL_1028
638 DECFSZ TEMPYY,F
639 GOTO LABEL_1027
640 NOP
641
642 ;FINISH AD-CONVERSION
643 BANK0
644 LABEL_1029
645 BTFSZ ADCON0,2
646 GOTO LABEL_1029
647 ;SET VARIABLE US WITH SFR
648 BANK0
649 MOVF ADRESH,W
650 MOVWF ADHBYTE
651 ;SET VARIABLE US WITH SFR
652 BANK1
653 MOVF ADRESL,W
654 BANK0
655 MOVWF ADLBYTE
656 ;SET VARIABLE U16 WITH VARIABLE US
657 MOVF ADLBYTE,W
658 MOVWF AD_RESULT
659 CLRF AD_RESULT_H
660 ;SET 16-BIT REGISTER HIGH BYTE THROUGH 8-BIT REGISTER
661 MOVF ADHBYTE,W
662 MOVWF AD_RESULT_H
663
664 INCF DATA_POINTER,F
665 BTFSZ STATUS,Z
666 INCF DATA_POINTER_H,F
667
668 ;SET INDIRECT VARIABLE WITH VARIABLE US
669 ;SET U16 POINTER
670 BCF STATUS,IRP
671 BANK0
672 BTFSZ DATA_POINTER_H,0
673 BSF STATUS,IRP
674 BANK0
675 MOVF DATA_POINTER,W
676 MOVWF FSR
677 MOVF ADHBYTE,W
678 MOVWF INDF
679
680 INCF DATA_POINTER,F
681 BTFSZ STATUS,Z
682 INCF DATA_POINTER_H,F
683
684 ;SET INDIRECT VARIABLE WITH VARIABLE US
685 ;SET U16 POINTER
686 BCF STATUS,IRP
687 BANK0
688 BTFSZ DATA_POINTER_H,0
689 BSF STATUS,IRP
690 BANK0
691 MOVF DATA_POINTER,W
692 MOVWF FSR
693 MOVF ADLBYTE,W
694 MOVWF INDF
695
696 ;SELECT AD-CHANNEL
697 BCF ADCON0,3
698 BCF ADCON0,4
699 BCF ADCON0,5
700 CALL LABEL_AQUISITION
701
702 ;START AD-CONVERSION
703 BANK0
704 BSF ADCON0,2
705
706 ;WAIT
707 MOVLW 0X1D
708 MOVWF TEMPYY
709 LABEL_1030
710 MOVLW 0XAC
711 MOVWF IDX16
712 LABEL_1031
713 DECFSZ IDX16,F
714 GOTO LABEL_1031
715 DECFSZ TEMPYY,F
716 GOTO LABEL_1030
717 NOP
718
719 ;FINISH AD-CONVERSION
720 BANK0
721 LABEL_1032
722 BTFSZ ADCON0,2
723 GOTO LABEL_1032
724 ;SET VARIABLE US WITH SFR
725 BANK0
726 MOVF ADRESH,W
727 MOVWF ADHBYTE
728 ;SET VARIABLE US WITH SFR
729 BANK1
730 MOVF ADRESL,W
731 BANK0
732 MOVWF ADLBYTE
733 ;SET VARIABLE U16 WITH VARIABLE US
734 MOVF ADLBYTE,W
735 MOVWF AD_RESULT
736 CLRF AD_RESULT_H
737 ;SET 16-BIT REGISTER HIGH BYTE THROUGH 8-BIT REGISTER

```

```

738 MOVF ADHBYTE,W
739 MOVWF AD_RESULT_H
740
741 INCF DATA_POINTER,F
742 BTFSC STATUS,Z
743 INCF DATA_POINTER_H,F
744
745 ;SET INDIRECT VARIABLE WITH VARIABLE US
746 ;SET U16 POINTER
747 BCF STATUS,IRP
748 BANK0
749 BTFSC DATA_POINTER_H,0
750 BSF STATUS,IRP
751 BANK0
752 MOVF DATA_POINTER,W
753 MOVWF FSR
754 MOVF ADHBYTE,W
755 MOVWF INDF
756
757 INCF DATA_POINTER,F
758 BTFSC STATUS,Z
759 INCF DATA_POINTER_H,F
760
761 ;SET INDIRECT VARIABLE WITH VARIABLE US
762 ;SET U16 POINTER
763 BCF STATUS,IRP
764 BANK0
765 BTFSC DATA_POINTER_H,0
766 BSF STATUS,IRP
767 BANK0
768 MOVF DATA_POINTER,W
769 MOVWF FSR
770 MOVF ADLBYTE,W
771 MOVWF INDF
772
773 ;SET VARIABLE U16 WITH CONSTANT
774 MOVLW 0xA6
775 MOVWF DATA_POINTER
776 MOVLW 0x0
777 MOVWF DATA_POINTER_H
778
779 ;BEGIN OF IF-STRUCTURE
780 ;COMPARE-OPERATION
781 ;SET VARIABLE US WITH INDIRECT VARIABLE
782 ;SET U16 POINTER
783 BCF STATUS,IRP
784 BTFSC DATA_POINTER_H,0
785 BSF STATUS,IRP
786 BANK0
787 MOVF DATA_POINTER,W
788 MOVWF FSR
789 MOVF INDF,W
790 MOVWF TEMPX8
791 ;SET VARIABLE US WITH CONSTANT
792 MOVLW 0x0
793 MOVWF TEMPY8
794 CALL GRU8US
795 BANK0
796 MOVWF RESULTS
797 BTFSC RESULTS,0
798 GOTO LABEL_1033
799
800 ;SET VARIABLE US WITH CONSTANT
801 MOVLW 0x82
802 BANK0
803 MOVWF SLEEP_TIME
804
805 GOTO LABEL_1034
806 LABEL_1033
807
808 ;SET VARIABLE US WITH CONSTANT
809 MOVLW 0x1
810 BANK0
811 MOVWF SLEEP_TIME
812
813 LABEL_1034
814 ;END OF IF-STRUCTURE
815
816 ;SET VARIABLE US WITH CONSTANT
817 MOVLW 0x1
818 BANK0
819 MOVWF STATE
820
821 LABEL_1035
822 ;END OF IF-STRUCTURE
823
824 ;BEGIN OF IF-STRUCTURE
825 ;COMPARE-OPERATION
826 ;SET VARIABLE US WITH VARIABLE US
827 BANK0
828 MOVF STATE,W
829 MOVWF TEMPX8
830 ;SET VARIABLE US WITH CONSTANT
831 MOVLW 0x1
832 MOVWF TEMPY8
833 CALL EQU8US
834 BANK0
835 MOVWF RESULTS
836 BTFSC RESULTS,0
837 GOTO LABEL_1036
838
839 GOTO LABEL_1046
840 LABEL_1036
841
842 ;SET SINGLE OUPUT PIN
843 BANK0
844 BSF PORTB,4
845
846 ;WAIT
847 MOVLW 0x63
848 MOVWF TEMPYY
849 LABEL_1037
850 MOVLW 0xA5
851 MOVWF IDI16
852 LABEL_1038
853 DECFSZ IDI16,F
854 GOTO LABEL_1038
855 DECFSZ TEMPYY,F
856 GOTO LABEL_1037

```

```

857      NOP
858      NOP
859
860      ;SET SINGLE OUPUT PIN
861      BANK0
862      BCF PORTB,4
863
864      ;CONFIGURE SINGLE PIN
865      BANK1
866      BCF TRISB,0
867
868      ;SET SINGLE OUPUT PIN
869      BANK0
870      BSF PORTB,0
871
872      ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
873      LABEL_1039
874      BANK0
875      BTFSC PORTB,1
876      GOTO LABEL_1040
877
878      GOTO LABEL_1039
879      LABEL_1040
880      ;END OF FOR/WHILE-LOOP
881
882      ;SET SINGLE OUPUT PIN
883      BANK0
884      BCF PORTB,0
885
886      ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
887      LABEL_1041
888      BANK0
889      BTFSS PORTB,1
890      GOTO LABEL_1042
891
892      GOTO LABEL_1041
893      LABEL_1042
894      ;END OF FOR/WHILE-LOOP
895
896      ;CONFIGURE SINGLE PIN
897      BANK1
898      BCF TRISB,3
899
900      CALL LABEL_SEND_0_RF
901
902      ;CONFIGURE SINGLE PIN
903      BANK1
904      BSF TRISB,0
905
906      ;CONFIGURE SINGLE PIN
907      BSF TRISB,3
908
909      CALL LABEL_INIT
910
911      ;WAIT
912      BANK0
913      MOVLW 0x6D
914      MOVWF TEMPYY
915      LABEL_1043
916      MOVLW 0x3
917      MOVWF IDX16
918      LABEL_1044
919      MOVLW 0xC7
920      MOVWF IDX16_H
921      LABEL_1045
922      DECFSZ IDX16_H,F
923      GOTO LABEL_1045
924      DECFSZ IDX16,F
925      GOTO LABEL_1044
926      DECFSZ TEMPYY,F
927      GOTO LABEL_1043
928      NOP
929      NOP
930
931      ;SET SINGLE OUPUT PIN
932      BANK0
933      BCF PORTA,3
934
935      ;SET VARIABLE US WITH CONSTANT
936      MOVLW 0x0
937      MOVWF STATE
938
939      LABEL_1046
940      ;END OF IF-STRUCTURE
941
942      ;BEGIN OF IF-STRUCTURE
943      ;COMPARE-OPERATION
944      ;SET VARIABLE US WITH VARIABLE US
945      BANK0
946      MOVF STATE,W
947      MOVWF TEMPX8
948      ;SET VARIABLE US WITH CONSTANT
949      MOVLW 0x0
950      MOVWF TEMPY8
951      CALL EQU8U8
952      BANK0
953      MOVWF RESULTS8
954      BTFSC RESULTS8,0
955      GOTO LABEL_1047
956
957      GOTO LABEL_1050
958      LABEL_1047
959
960      BANK0
961      INCF WDT_EVENTS,F
962
963      ;BEGIN OF IF-STRUCTURE
964      ;COMPARE-OPERATION
965      ;SET VARIABLE US WITH VARIABLE US
966      MOVF WDT_EVENTS,W
967      MOVWF TEMPX8
968      ;SET VARIABLE US WITH VARIABLE US
969      MOVF SLEEP_TIME,W
970      MOVWF TEMPY8
971      CALL GRU8U8
972      BANK0
973      MOVWF RESULTS8
974      BTFSC RESULTS8,0
975      GOTO LABEL_1048

```

```

976
977
978             BANKO
979             CLRF WDT_EVENTS
980
981             ;SET VARIABLE US WITH CONSTANT
982             MOVLW 0X6
983             MOVWF STATE
984
985             GOTO LABEL_1049
986             LABEL_1048
987
988             LABEL_1049
989             ;END OF IF-STRUCTURE
990
991             SLEEP
992             CLRWDI
993
994 LABEL_1050
995 ;END OF IF-STRUCTURE
996
997             GOTO LABEL_0
998 LABEL_1051
999
1000            GOTO LABEL_1051
1001
1002            ***
1003            ;*****SUBROUTINES*****
1004 LABEL_AQUISITION
1005
1006            ;WAIT
1007            BANKO
1008            MOVLW 0X6
1009            MOVWF TEMPYY
1010 LABEL_1002
1011            DECFSSZ TEMPYY,F
1012            GOTO LABEL_1002
1013            NOP
1014            NOP
1015
1016            RETURN
1017 LABEL_SEND_RF_DATA_BIT
1018
1019            ;BEGIN OF IF-STRUCTURE (DEPENDING ON BIT/PIN)
1020            BANKO
1021            BTFSS RF_OUT_DATA,7
1022            GOTO LABEL_1003
1023
1024            ;SET SINGLE OUPUT PIN
1025            BANKO
1026            BSF PORTB,3
1027
1028            GOTO LABEL_1004
1029 LABEL_1003
1030
1031            ;SET SINGLE OUPUT PIN
1032            BANKO
1033            BCF PORTB,3
1034
1035 LABEL_1004
1036 ;END OF IF-STRUCTURE
1037
1038            ;OPERATION
1039            ;SET VARIABLE US WITH VARIABLE US
1040            BANKO
1041            MOVF RF_OUT_DATA,W
1042            MOVWF TEMPY8
1043            CALL SHLW8
1044            ;SET VARIABLE US WITH VARIABLE US
1045            BANKO
1046            MOVF TEMPY8,W
1047            MOVWF RF_OUT_DATA
1048
1049            ;SET SINGLE OUPUT PIN
1050            BSF PORTB,0
1051
1052            ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
1053 LABEL_1005
1054            BANKO
1055            BTFSC PORTB,1
1056            GOTO LABEL_1006
1057
1058            GOTO LABEL_1005
1059 LABEL_1006
1060 ;END OF FOR/WHILE-LOOP
1061
1062            ;SET SINGLE OUPUT PIN
1063            BANKO
1064            BCF PORTB,0
1065
1066            ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
1067 LABEL_1007
1068            BANKO
1069            BTFSS PORTB,1
1070            GOTO LABEL_1008
1071
1072            GOTO LABEL_1007
1073 LABEL_1008
1074 ;END OF FOR/WHILE-LOOP
1075
1076            RETURN
1077 LABEL_SEND_0_RF
1078
1079            ;BEGIN OF FOR-LOOP
1080            ;SET VARIABLE US WITH VARIABLE US
1081            BANKO
1082            MOVF RF_FRAME_LENGTH,W
1083            MOVWF BYTE_COUNTER
1084            INCF BYTE_COUNTER,F
1085 LABEL_1009
1086            BANKO
1087            DECF BYTE_COUNTER,F
1088            BTFSC STATUS,Z
1089            GOTO LABEL_1010
1090
1091            ;SET VARIABLE US WITH INDIRECT VARIABLE
1092            ;SET U16 POINTER
1093
1094

```

```

1095         BCF STATUS,IRP
1096         BANK0
1097         BTFSB POINTER_RF_OUT_H,0
1098         BSF STATUS,IRP
1099         BANK0
1100         MOVWF POINTER_RF_OUT,W
1101         MOVWF FSR
1102         MOVF INDF,W
1103         MOVWF RF_OUT_DATA
1104
1105         INCF POINTER_RF_OUT,F
1106         BTFSB STATUS,Z
1107         INCF POINTER_RF_OUT_H,F
1108
1109         ;BEGIN OF FOR-LOOP
1110         ;SET VARIABLE U8 WITH CONSTANT
1111         MOVLW 0X8
1112         BANK0
1113         MOVWF BIT_COUNTER
1114         INCF BIT_COUNTER,F
1115         LABEL_1011
1116         BANK0
1117         DECF BIT_COUNTER,F
1118         BTFSB STATUS,Z
1119         GOTO LABEL_1012
1120
1121                 CALL LABEL_SEND_RF_DATA_BIT
1122
1123         GOTO LABEL_1011
1124         LABEL_1012
1125         ;END OF FOR/WHILE-LOOP
1126
1127         GOTO LABEL_1009
1128 LABEL_1010
1129 ;END OF FOR/WHILE-LOOP
1130
1131         RETURN
1132
1133 LABEL_INIT
1134
1135         ;SET VARIABLE U16 WITH CONSTANT
1136         MOVLW 0XA0
1137         BANK0
1138         MOVWF POINTER_RF_OUT
1139         MOVLW 0X0
1140         MOVWF POINTER_RF_OUT_H
1141
1142         ;SET VARIABLE U16 WITH CONSTANT
1143         MOVLW 0X20
1144         MOVWF POINTER_RF_IN
1145         MOVLW 0X1
1146         MOVWF POINTER_RF_IN_H
1147
1148         CLRF RF_PACKET_COUNTER
1149
1150         ;SET VARIABLE U8 WITH CONSTANT
1151         MOVLW 0X8
1152         MOVWF RF_FRAME_LENGTH
1153
1154         CLRF RF_FRAME_INDEX
1155
1156         ;CONFIGURE SINGLE PIN
1157         BANK1
1158         BSF TRISB,0
1159
1160         ;CONFIGURE SINGLE PIN
1161         BSF TRISB,1
1162
1163         ;CONFIGURE SINGLE PIN
1164         BSF TRISB,3
1165
1166         RETURN
1167
1168
1169
1170         END

```

Part IV

433MHz receiver and 4-digit display

8 Device schematics

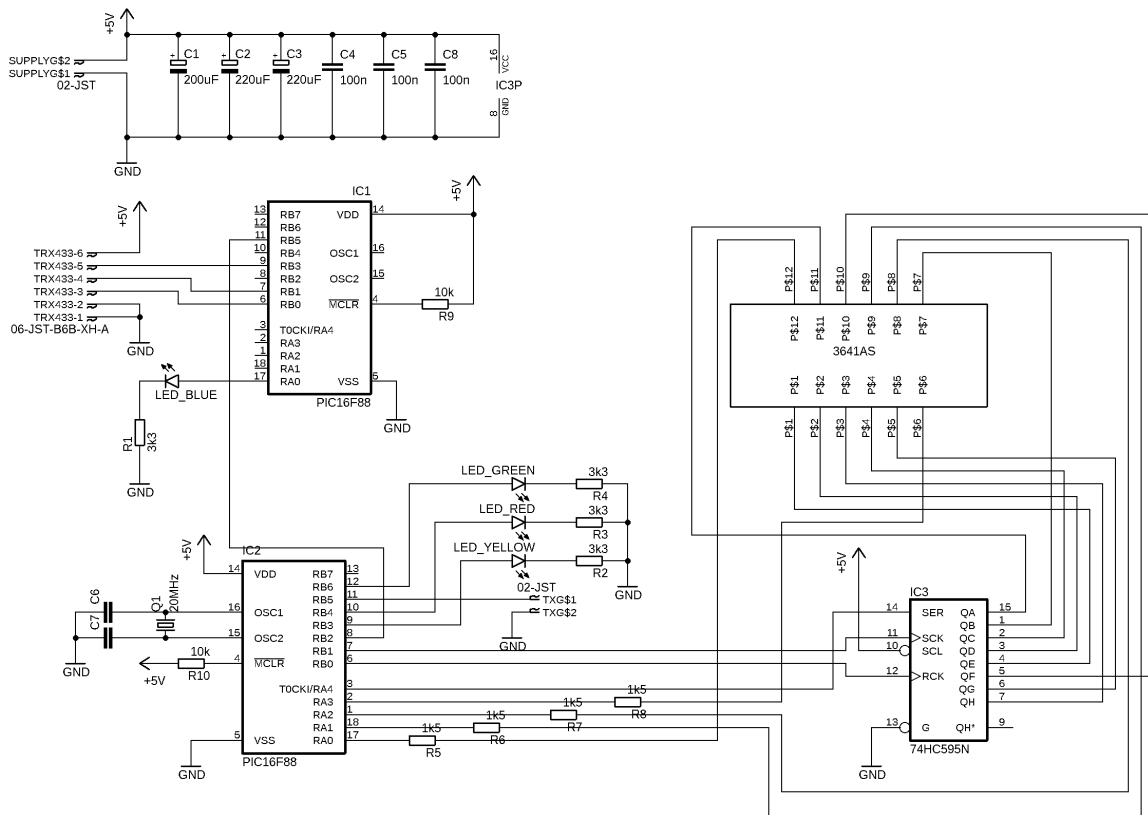


Figure 17: Receiver device schematics.

The device consists of two PIC16F88 microcontrollers (MCUs). The first MCU (IC1), operating at 8 MHz using its internal oscillator, manages communication with the TRX433 module. Upon receiving an RF packet, the module signals this by setting the DATA line HIGH, as previously described. IC1 then transmits the data to the second MCU (IC2) via a 9600 baud UART connection. To facilitate UART reception on IC2, IC1 precedes each packet with a three-byte header (0x55, 0xFF, 0x00) and briefly flashes a blue LED.

IC2 controls a four-digit 7-segment display using an interrupt-driven state machine, as detailed earlier. If the sender's battery power is sufficient, the green LED illuminates; otherwise, the red LED lights up. IC2 expects data from IC1 within 6 minutes. Upon receiving data, the yellow LED turns on; if no data is received, the yellow LED remains off. For system stability, IC2 is clocked by a 20 MHz crystal oscillator.

9 PICLab Program (RX_1.0) for IC1

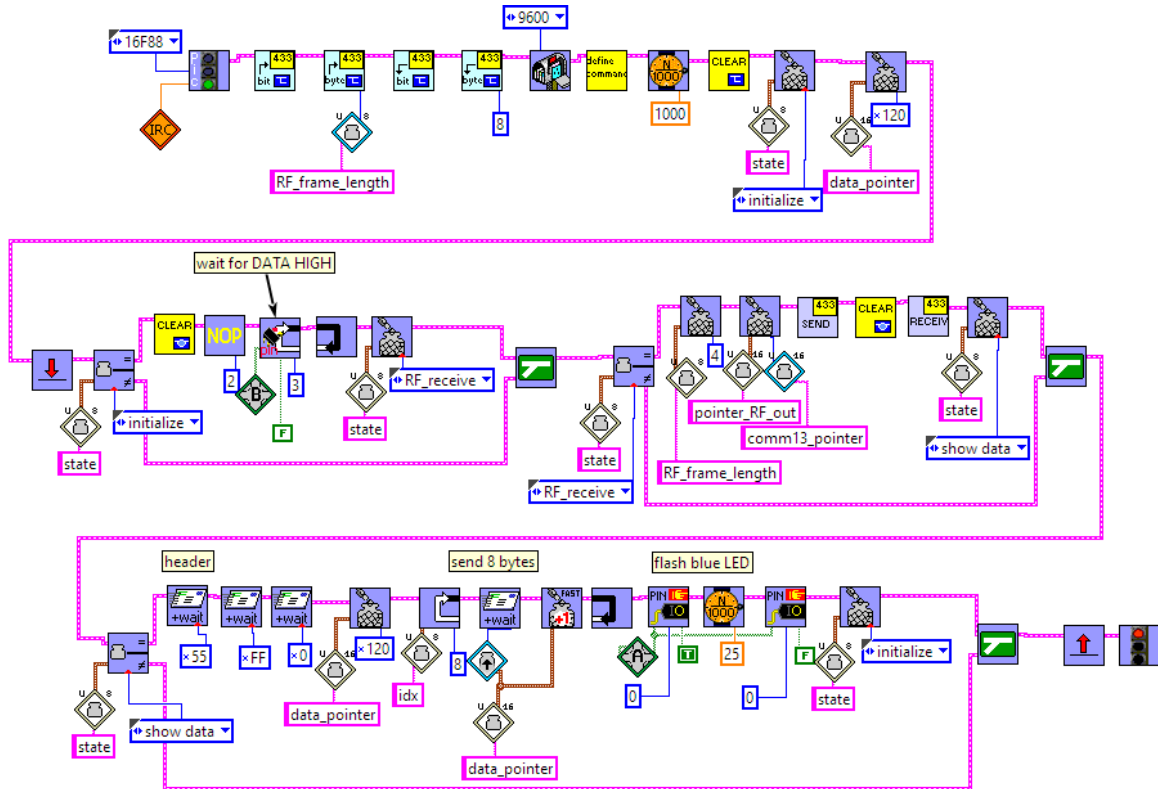


Figure 18: PICLAB main RX program for IC1 (filename: *test_433_RX.vi*).

10 Assembler code (RX_1.0) for IC1

```

1  ;-----
2  ;                               Assembler code for test_433_RX.vi created with PICLab
3  ;-----
4
5  LIST    p=16F88
6
7  #include "P16F88.INC" ; Include header file
8
9  __CONFIG _CONFIG1, _BODEN_OFF&_CP_OFF&_PWRTE_ON&_WDT_OFF&_LVP_OFF&_MCLR_ON&_INTRC_IO
10
11 ;*****Variable definitions*****
12
13 TEMPPORT8 EQU 0X20
14 TEMPX8 EQU 0X21
15 TEMPY8 EQU 0X22
16 RESULT8 EQU 0X23
17 TEMPX16 EQU 0X24
18 TEMPX16_H EQU 0X25
19 TEMPY16 EQU 0X26
20 TEMPY16_H EQU 0X27
21 RESULT16 EQU 0X28
22 RESULT16_H EQU 0X29
23 IDX16 EQU 0X2A
24 IDX16_H EQU 0X2B
25 TEMPYY EQU 0X2C
26 OP_SIGN8 EQU 0X2D
27 ADHBYTE EQU 0X2E
28 ADLBYTE EQU 0X2F
29 AD_RESULT EQU 0X30
30 AD_RESULT_H EQU 0X31
31 ISR_TEMPPORT8 EQU 0X32
32 ISR_TEMPX8 EQU 0X33
33 ISR_TEMPY8 EQU 0X34
34 ISR_RESULT8 EQU 0X35
35 ISR_TEMPX16 EQU 0X36
36 ISR_TEMPX16_H EQU 0X37
37 ISR_TEMPY16 EQU 0X38
38 ISR_TEMPY16_H EQU 0X39
39 ISR_RESULT16 EQU 0X3A
40 ISR_RESULT16_H EQU 0X3B
41 ISR_IDX16 EQU 0X3C
42 ISR_IDX16_H EQU 0X3D
43 ISR_TEMPYY EQU 0X3E
44 ISR_OP_SIGN8 EQU 0X3F
45 ISR_STATUS EQU 0X40
46 ISR_W EQU 0X41
47 ISR_FSR EQU 0X42
48 DATA_POINTER EQU 0X43
49 DATA_POINTER_H EQU 0X44

```

```

50 STATE EQU 0X45
51 RF_FRAME_LENGTH EQU 0X46
52 COMM13_POINTER EQU 0X47
53 COMM13_POINTER_H EQU 0X48
54 RF_OUT_DATA EQU 0X49
55 POINTER_RF_OUT EQU 0X4A
56 POINTER_RF_OUT_H EQU 0X4B
57 IDX EQU 0X4C
58 BIT_COUNTER EQU 0X4D
59 BYTE_COUNTER EQU 0X4E
60 RF_IN_DATA EQU 0X4F
61 POINTER_RF_IN EQU 0X50
62 POINTER_RF_IN_H EQU 0X51
63 COMM6_POINTER EQU 0X52
64 COMM6_POINTER_H EQU 0X53
65 COMM7_POINTER EQU 0X54
66 COMM7_POINTER_H EQU 0X55
67 RF_FRAME_INDEX EQU 0X56
68 RF_PACKET_COUNTER EQU 0X57
69
70 ;*****Makro definitions and definitions of used operations*****
71
72 BANK0 MACRO
73     BCF STATUS,RPO
74     BCF STATUS,RP1
75     ENDM
76 BANK1 MACRO
77     BSF STATUS,RPO
78     BCF STATUS,RP1
79     ENDM
80 BANK2 MACRO
81     BCF STATUS,RPO
82     BSF STATUS,RP1
83     ENDM
84 BANK3 MACRO
85     BSF STATUS,RPO
86     BSF STATUS,RP1
87     ENDM
88
89     GOTO START
90     ORG 0X4
91
92 SHLV8
93     BCF STATUS,C
94     RLF TEMPY8,F
95     RETURN
96
97 EQU8U8
98     MOVF TEMPX8,W
99     SUBWF TEMPY8
100    BTFSS STATUS,Z ;SKIP IF ZERO ->IDENTICAL
101    RETLW 0
102    RETLW 1
103
104 ;*****BEGIN OF MAIN PROGRAM*****
105
106 START
107
108 ; INITIALIZE PORT A
109 BANK0
110 CLRF PORTA
111 BANK1
112 CLRF ANSEL
113 MOVLW 0X07
114 MOVWF CMCON
115 MOVLW 0X0
116 MOVWF TRISA
117
118 ; INITIALIZE PORT B
119 BANK0
120 CLRF PORTB
121 BANK1
122 MOVLW 0X0
123 MOVWF TRISB
124
125 ; CONFIGURE INTERNAL OSCILLATOR
126 MOVLW 0XFC
127 MOVWF DSCCON
128
129 ; CONFIGURE UART
130 BSF TRISB,2
131 BSF TRISB,5
132 MOVLW 0XC
133 MOVWF SPBRG
134 BCF TXSTA,BRGH
135 BCF TXSTA,SYNC
136 BANK0
137 BSF RCSTA,SPEN
138 BCF RCSTA,RX9
139 BSF RCSTA,CREN
140 BANK1
141 BCF TXSTA,TX9
142 BSF TXSTA,TXEN
143
144 ; DEFINE COMM13_POINTER
145
146 ;SET VARIABLE U16 WITH CONSTANT
147 MOVLW 0X10
148 BANK0
149 MOVWF COMM13_POINTER
150 MOVLW 0X1
151 MOVWF COMM13_POINTER_H
152
153 ;SET INDIRECT VARIABLE WITH CONSTANT
154 ;SET U16_POINTER
155 BCF STATUS,IRP
156 BTFSC COMM13_POINTER_H,0
157 BSF STATUS,IRP
158 BANK0
159 MOVF COMM13_POINTER,W
160 MOVWF FSR
161 MOVLW 0X0
162 MOVWF INDF
163
164 INCF COMM13_POINTER,F
165 BTFSC STATUS,Z
166 INCF COMM13_POINTER_H,F
167
168 ;SET INDIRECT VARIABLE WITH CONSTANT

```

```

169 ;SET U16 POINTER
170 BCF STATUS,IRP
171 BANKO
172 BTFSC COMM13_POINTER_H,0
173 BSF STATUS,IRP
174 BANKO
175 MOVF COMM13_POINTER,W
176 MOVWF FSR
177 MOVLW OXD
178 MOVWF INDF
179
180 INCF COMM13_POINTER,F
181 BTFSC STATUS,Z
182 INCF COMM13_POINTER_H,F
183
184 ;SET INDIRECT VARIABLE WITH CONSTANT
185 ;SET U16 POINTER
186 BCF STATUS,IRP
187 BANKO
188 BTFSC COMM13_POINTER_H,0
189 BSF STATUS,IRP
190 BANKO
191 MOVF COMM13_POINTER,W
192 MOVWF FSR
193 MOVLW OX0
194 MOVWF INDF
195
196 INCF COMM13_POINTER,F
197 BTFSC STATUS,Z
198 INCF COMM13_POINTER_H,F
199
200 ;SET INDIRECT VARIABLE WITH CONSTANT
201 ;SET U16 POINTER
202 BCF STATUS,IRP
203 BANKO
204 BTFSC COMM13_POINTER_H,0
205 BSF STATUS,IRP
206 BANKO
207 MOVF COMM13_POINTER,W
208 MOVWF FSR
209 MOVLW OX0
210 MOVWF INDF
211
212 ;SET VARIABLE U16 WITH CONSTANT
213 MOVLW OX10
214 MOVWF COMM13_POINTER
215 MOVLW OX1
216 MOVWF COMM13_POINTER_H
217
218 ; DEFINE COMM6_POINTER
219
220 ;SET VARIABLE U16 WITH CONSTANT
221 MOVLW OX14
222 MOVWF COMM6_POINTER
223 MOVLW OX1
224 MOVWF COMM6_POINTER_H
225
226 ;SET INDIRECT VARIABLE WITH CONSTANT
227 ;SET U16 POINTER
228 BCF STATUS,IRP
229 BTFSC COMM6_POINTER_H,0
230 BSF STATUS,IRP
231 BANKO
232 MOVF COMM6_POINTER,W
233 MOVWF FSR
234 MOVLW OX0
235 MOVWF INDF
236
237 INCF COMM6_POINTER,F
238 BTFSC STATUS,Z
239 INCF COMM6_POINTER_H,F
240
241 ;SET INDIRECT VARIABLE WITH CONSTANT
242 ;SET U16 POINTER
243 BCF STATUS,IRP
244 BANKO
245 BTFSC COMM6_POINTER_H,0
246 BSF STATUS,IRP
247 BANKO
248 MOVF COMM6_POINTER,W
249 MOVWF FSR
250 MOVLW OX6
251 MOVWF INDF
252
253 INCF COMM6_POINTER,F
254 BTFSC STATUS,Z
255 INCF COMM6_POINTER_H,F
256
257 ;SET INDIRECT VARIABLE WITH CONSTANT
258 ;SET U16 POINTER
259 BCF STATUS,IRP
260 BANKO
261 BTFSC COMM6_POINTER_H,0
262 BSF STATUS,IRP
263 BANKO
264 MOVF COMM6_POINTER,W
265 MOVWF FSR
266 MOVLW OX6F
267 MOVWF INDF
268
269 INCF COMM6_POINTER,F
270 BTFSC STATUS,Z
271 INCF COMM6_POINTER_H,F
272
273 ;SET INDIRECT VARIABLE WITH CONSTANT
274 ;SET U16 POINTER
275 BCF STATUS,IRP
276 BANKO
277 BTFSC COMM6_POINTER_H,0
278 BSF STATUS,IRP
279 BANKO
280 MOVF COMM6_POINTER,W
281 MOVWF FSR
282 MOVLW OX0
283 MOVWF INDF
284
285 ;SET VARIABLE U16 WITH CONSTANT
286 MOVLW OX14
287 MOVWF COMM6_POINTER

```

```

288 MOVW 0X1
289 MOVWF COMM6_POINTER_H
290
291 ; DEFINE COMM7_POINTER
292
293 ;SET VARIABLE U16 WITH CONSTANT
294 MOVW 0X18
295 MOVWF COMM7_POINTER
296 MOVW 0X1
297 MOVWF COMM7_POINTER_H
298
299 ;SET INDIRECT VARIABLE WITH CONSTANT
300 ;SET U16_POINTER
301 BCF STATUS,IRP
302 BTFSZ COMM7_POINTER_H,0
303 BSF STATUS,IRP
304 BANK0
305 MOVF COMM7_POINTER,W
306 MOVWF FSR
307 MOVW 0X0
308 MOVWF INDF
309
310 INCF COMM7_POINTER,F
311 BTFSZ STATUS,Z
312 INCF COMM7_POINTER_H,F
313
314 ;SET INDIRECT VARIABLE WITH CONSTANT
315 ;SET U16_POINTER
316 BCF STATUS,IRP
317 BANK0
318 BTFSZ COMM7_POINTER_H,0
319 BSF STATUS,IRP
320 BANK0
321 MOVF COMM7_POINTER,W
322 MOVWF FSR
323 MOVW 0X7
324 MOVWF INDF
325
326 INCF COMM7_POINTER,F
327 BTFSZ STATUS,Z
328 INCF COMM7_POINTER_H,F
329
330 ;SET INDIRECT VARIABLE WITH CONSTANT
331 ;SET U16_POINTER
332 BCF STATUS,IRP
333 BANK0
334 BTFSZ COMM7_POINTER_H,0
335 BSF STATUS,IRP
336 BANK0
337 MOVF COMM7_POINTER,W
338 MOVWF FSR
339 MOVW 0X3
340 MOVWF INDF
341
342 INCF COMM7_POINTER,F
343 BTFSZ STATUS,Z
344 INCF COMM7_POINTER_H,F
345
346 ;SET INDIRECT VARIABLE WITH CONSTANT
347 ;SET U16_POINTER
348 BCF STATUS,IRP
349 BANK0
350 BTFSZ COMM7_POINTER_H,0
351 BSF STATUS,IRP
352 BANK0
353 MOVF COMM7_POINTER,W
354 MOVWF FSR
355 MOVW 0X0
356 MOVWF INDF
357
358 ;SET VARIABLE U16 WITH CONSTANT
359 MOVW 0X18
360 MOVWF COMM7_POINTER
361 MOVW 0X1
362 MOVWF COMM7_POINTER_H
363
364 ;WAIT
365 MOVW 0XD0
366 MOVWF TEMPYY
367 LABEL_1022
368 MOVW 0X13
369 MOVWF IDX16
370 LABEL_1023
371 MOVW 0XA7
372 MOVWF IDX16_H
373 LABEL_1024
374 DECFSZ IDX16_H,F
375 GOTO LABEL_1024
376 DECFSZ IDX16,F
377 GOTO LABEL_1023
378 DECFSZ TEMPYY,F
379 GOTO LABEL_1022
380 NOP
381 NOP
382 NOP
383 NOP
384 NOP
385
386 ;SET VARIABLE U8 WITH CONSTANT
387 MOVW 0X6
388 BANK0
389 MOVWF STATE
390
391 ;SET VARIABLE U16 WITH CONSTANT
392 MOVW 0X20
393 MOVWF DATA_POINTER
394 MOVW 0X1
395 MOVWF DATA_POINTER_H
396
397 LABEL_0
398
399 ;BEGIN OF IF-STRUCTURE
400 ;COMPARE-OPERATION
401 ;SET VARIABLE U8 WITH VARIABLE U8
402 BANK0
403 MOVF STATE,W
404 MOVWF TEMPX8
405 ;SET VARIABLE U8 WITH CONSTANT
406 MOVW 0X6

```

```

407 MOVWF TEMPY8
408 CALL EQU8U8
409 BANK0
410 MOVWF RESULTS8
411 BTFSC RESULTS8,0
412 GOTO LABEL_1025
413
414 GOTO LABEL_1028
415 LABEL_1025
416
417 CALL LABEL_INIT
418
419 NOP
420 NOP
421
422 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
423 LABEL_1026
424 BANK0
425 BTFSC PORTB,3
426 GOTO LABEL_1027
427
428 GOTO LABEL_1026
429 LABEL_1027
430 ;END OF FOR/WHILE-LOOP
431
432 ;SET VARIABLE U8 WITH CONSTANT
433 MOVLW 0X2
434 BANK0
435 MOVWF STATE
436
437 LABEL_1028
438 ;END OF IF-STRUCTURE
439
440 ;BEGIN OF IF-STRUCTURE
441 ;COMPARE-OPERATION
442 ;SET VARIABLE U8 WITH VARIABLE U8
443 BANK0
444 MOVF STATE,W
445 MOVWF TEMPX8
446 ;SET VARIABLE U8 WITH CONSTANT
447 MOVLW 0X2
448 MOVWF TEMPY8
449 CALL EQU8U8
450 BANK0
451 MOVWF RESULTS8
452 BTFSC RESULTS8,0
453 GOTO LABEL_1029
454
455 GOTO LABEL_1040
456 LABEL_1029
457
458 ;SET VARIABLE U8 WITH CONSTANT
459 MOVLW 0X4
460 BANK0
461 MOVWF RF_FRAME_LENGTH
462
463 ;SET VARIABLE U16 WITH VARIABLE U16
464 MOVF COMM13_POINTER,W
465 MOVWF POINTER_RF_OUT
466 MOVF COMM13_POINTER_H,W
467 MOVWF POINTER_RF_OUT_H
468
469 ;CONFIGURE SINGLE PIN
470 BANK1
471 BCF TRISB,0
472
473 ;SET SINGLE OUPUT PIN
474 BANK0
475 BSF PORTB,0
476
477 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
478 LABEL_1030
479 BANK0
480 BTFSC PORTB,1
481 GOTO LABEL_1031
482
483 GOTO LABEL_1030
484 LABEL_1031
485 ;END OF FOR/WHILE-LOOP
486
487 ;SET SINGLE OUPUT PIN
488 BANK0
489 BCF PORTB,0
490
491 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
492 LABEL_1032
493 BANK0
494 BTFSS PORTB,1
495 GOTO LABEL_1033
496
497 GOTO LABEL_1032
498 LABEL_1033
499 ;END OF FOR/WHILE-LOOP
500
501 ;CONFIGURE SINGLE PIN
502 BANK1
503 BCF TRISB,3
504
505 CALL LABEL_SEND_0_RF
506
507 ;CONFIGURE SINGLE PIN
508 BANK1
509 BSF TRISB,0
510
511 ;CONFIGURE SINGLE PIN
512 BSF TRISB,3
513
514 CALL LABEL_INIT
515
516 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
517 LABEL_1034
518 BANK0
519 BTFSC PORTB,0
520 GOTO LABEL_1035
521
522 GOTO LABEL_1034
523 LABEL_1035
524 ;END OF FOR/WHILE-LOOP
525

```

```

526         ;CONFIGURE SINGLE PIN
527         BANK1
528         BCF TRISB,1
529
530         ;SET SINGLE OUPUT PIN
531         BANK0
532         BSF PORTB,1
533
534         ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
535         LABEL_1036
536         BANK0
537         BTFSS PORTB,0
538         GOTO LABEL_1037
539
540         GOTO LABEL_1036
541         LABEL_1037
542         ;END OF FOR/WHILE-LOOP
543
544         ;SET SINGLE OUPUT PIN
545         BANK0
546         BCF PORTB,1
547
548         ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
549         LABEL_1038
550         BANK0
551         BTFSS PORTB,0
552         GOTO LABEL_1039
553
554         GOTO LABEL_1038
555         LABEL_1039
556         ;END OF FOR/WHILE-LOOP
557
558         CALL LABEL_RECEIVE @_RF
559
560         ;CONFIGURE SINGLE PIN
561         BANK1
562         BSF TRISB,1
563
564         ;SET VARIABLE U8 WITH CONSTANT
565         MOVLW 0X5
566         BANK0
567         MOVWF STATE
568
569         LABEL_1040
570         ;END OF IF-STRUCTURE
571
572         ;BEGIN OF IF-STRUCTURE
573         ;COMPARE-OPERATION
574         ;SET VARIABLE U8 WITH VARIABLE U8
575         BANK0
576         MOVF STATE,W
577         MOVWF TEMPX8
578         ;SET VARIABLE U8 WITH CONSTANT
579         MOVLW 0X5
580         MOVWF TEMPY8
581         CALL EQU8U8
582         BANK0
583         MOVWF RESULT8
584         BTFSC RESULT8,0
585         GOTO LABEL_1041
586
587         GOTO LABEL_1054
588         LABEL_1041
589
590         ;SEND MAIL
591         BANK0
592         LABEL_1043
593         BTFSS PIR1,TXIF
594         GOTO LABEL_1043
595         ;SET SFR WITH CONSTANT
596         MOVLW 0X55
597         BANK0
598         MOVWF TXREG
599
600
601         LABEL_1042
602
603         BTFSS PIR1,TXIF
604
605         GOTO LABEL_1042
606
607         ;SEND MAIL
608         BANK0
609         LABEL_1045
610         BTFSS PIR1,TXIF
611         GOTO LABEL_1045
612         ;SET SFR WITH CONSTANT
613         MOVLW 0XFF
614         BANK0
615         MOVWF TXREG
616
617
618         LABEL_1044
619
620         BTFSS PIR1,TXIF
621
622         GOTO LABEL_1044
623
624         ;SEND MAIL
625         BANK0
626         LABEL_1047
627         BTFSS PIR1,TXIF
628         GOTO LABEL_1047
629         ;SET SFR WITH CONSTANT
630         MOVLW 0X0
631         BANK0
632         MOVWF TXREG
633
634
635         LABEL_1046
636
637         BTFSS PIR1,TXIF
638
639         GOTO LABEL_1046
640
641         ;SET VARIABLE U16 WITH CONSTANT
642         MOVLW 0X20
643         BANK0
644         MOVWF DATA_POINTER

```

```

645      MOVLW 0X1
646      MOVWF DATA_POINTER_H
647
648      ;BEGIN OF FOR-LOOP
649      ;SET VARIABLE US WITH CONSTANT
650      MOVLW 0X8
651      MOVWF IDX
652      INCF IDX,F
653      LABEL_1048
654      BANK0
655      DECF IDX,F
656      BTFSZ STATUS,Z
657      GOTO LABEL_1049
658
659      ;SEND MAIL
660      BANK0
661      LABEL_1051
662      BTFSZ PIR1,TXIF
663      GOTO LABEL_1051
664      ;SET SFR WITH INDIRECT VARIABLE
665      ;SET U16 POINTER
666      BCF STATUS,IRP
667      BANK0
668      BTFSZ DATA_POINTER_H,0
669      BSF STATUS,IRP
670      BANK0
671      MOVF DATA_POINTER,W
672      MOVWF FSR
673      MOVF INDF,W
674      MOVWF TXREG
675
676
677      LABEL_1050
678
679      BTFSZ PIR1,TXIF
680
681      GOTO LABEL_1050
682
683      BANK0
684      INCF DATA_POINTER,F
685      BTFSZ STATUS,Z
686      INCF DATA_POINTER_H,F
687
688      GOTO LABEL_1048
689      LABEL_1049
690      ;END OF FOR/WHILE-LOOP
691
692      ;SET SINGLE OUPUT PIN
693      BANK0
694      BSF PORTA,0
695
696      ;WAIT
697      MOVLW 0X63
698      MOVWF TEMPYY
699      LABEL_1052
700      MOVLW 0XA5
701      MOVWF IDX16
702      LABEL_1053
703      DECFSZ IDX16,F
704      GOTO LABEL_1053
705      DECFSZ TEMPYY,F
706      GOTO LABEL_1052
707      NOP
708      NOP
709
710      ;SET SINGLE OUPUT PIN
711      BANK0
712      BCF PORTA,0
713
714      ;SET VARIABLE US WITH CONSTANT
715      MOVLW 0X6
716      MOVWF STATE
717
718      LABEL_1054
719      ;END OF IF-STRUCTURE
720
721      GOTO LABEL_0
722
723      LABEL_1055
724
725      GOTO LABEL_1055
726
727      ;*****SUBROUTINES*****
728
729      LABEL_SEND_RF_DATA_BIT
730
731      ;BEGIN OF IF-STRUCTURE (DEPENDING ON BIT/PIN)
732      BANK0
733      BTFSZ RF_OUT_DATA,7
734      GOTO LABEL_1002
735
736      ;SET SINGLE OUPUT PIN
737      BANK0
738      BSF PORTB,3
739
740      GOTO LABEL_1003
741      LABEL_1002
742
743      ;SET SINGLE OUPUT PIN
744      BANK0
745      BCF PORTB,3
746
747      LABEL_1003
748      ;END OF IF-STRUCTURE
749
750      ;OPERATION
751      ;SET VARIABLE US WITH VARIABLE US
752      BANK0
753      MOVF RF_OUT_DATA,W
754      MOVWF TEMPY8
755      CALL SHLV8
756      ;SET VARIABLE US WITH VARIABLE US
757      BANK0
758      MOVF TEMPY8,W
759      MOVWF RF_OUT_DATA
760
761      ;SET SINGLE OUPUT PIN
762      BSF PORTB,0
763

```

```

764 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
765 LABEL_1004
766 BANKO
767 BTFSC PORTB,1
768 GOTO LABEL_1005
769
770 GOTO LABEL_1004
771 LABEL_1005
772 ;END OF FOR/WHILE-LOOP
773
774 ;SET SINGLE OUPUT PIN
775 BANKO
776 BCF PORTB,0
777
778 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
779 LABEL_1006
780 BANKO
781 BTFSS PORTB,1
782 GOTO LABEL_1007
783
784 GOTO LABEL_1006
785 LABEL_1007
786 ;END OF FOR/WHILE-LOOP
787
788 RETURN
789
790 LABEL_SEND_0_RF
791
792 ;BEGIN OF FOR-LOOP
793 ;SET VARIABLE US WITH VARIABLE US
794 BANKO
795 MOVF RF_FRAME_LENGTH,W
796 MOVWF BYTE_COUNTER
797 INCF BYTE_COUNTER,F
798 LABEL_1008
799 BANKO
800 DECF BYTE_COUNTER,F
801 BTFSC STATUS,Z
802 GOTO LABEL_1009
803
804 ;SET VARIABLE US WITH INDIRECT VARIABLE
805 ;SET U16 POINTER
806 BCF STATUS,IRP
807 BANKO
808 BTFSC POINTER_RF_OUT_H,0
809 BSF STATUS,IRP
810 BANKO
811 MOVF POINTER_RF_OUT,W
812 MOVWF FSR
813 MOVF INDF,W
814 MOVWF RF_OUT_DATA
815
816 INCF POINTER_RF_OUT,F
817 BTFSC STATUS,Z
818 INCF POINTER_RF_OUT_H,F
819
820 ;BEGIN OF FOR-LOOP
821 ;SET VARIABLE US WITH CONSTANT
822 MOVLW 0x8
823 BANKO
824 MOVWF BIT_COUNTER
825 INCF BIT_COUNTER,F
826 LABEL_1010
827 BANKO
828 DECF BIT_COUNTER,F
829 BTFSC STATUS,Z
830 GOTO LABEL_1011
831
832 CALL LABEL_SEND_RF_DATA_BIT
833
834 GOTO LABEL_1010
835 LABEL_1011
836 ;END OF FOR/WHILE-LOOP
837
838 GOTO LABEL_1008
839 LABEL_1009
840 ;END OF FOR/WHILE-LOOP
841
842 RETURN
843
844 LABEL_RECEIVE_RF_DATA_BIT
845
846 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
847 LABEL_1012
848 BANKO
849 BTFSC PORTB,0
850 GOTO LABEL_1013
851
852 GOTO LABEL_1012
853 LABEL_1013
854 ;END OF FOR/WHILE-LOOP
855
856 ;OPERATION
857 ;SET VARIABLE US WITH VARIABLE US
858 BANKO
859 MOVF RF_IN_DATA,W
860 MOVWF TEMPY8
861 CALL SHL8
862 ;SET VARIABLE US WITH VARIABLE US
863 BANKO
864 MOVF TEMPY8,W
865 MOVWF RF_IN_DATA
866
867 ;BEGIN OF IF-STRUCTURE (DEPENDING ON BIT/PIN)
868 BTFSS PORTB,3
869 GOTO LABEL_1014
870
871 BANKO
872 BSF RF_IN_DATA,0
873
874 GOTO LABEL_1015
875 LABEL_1014
876
877 BANKO
878 BCF RF_IN_DATA,0
879
880 LABEL_1015
881 ;END OF IF-STRUCTURE
882

```



```

883 ;SET SINGLE OUPUT PIN
884 BANK0
885 BSF PORTB,1
886
887 ;BEGIN OF WHILE-LOOP (POLLING BIT/PIN)
888 LABEL_1016
889 BANK0
890 BTFSS PORTB,0
891 GOTO LABEL_1017
892
893 GOTO LABEL_1016
894 LABEL_1017
895 ;END OF FOR/WHILE-LOOP
896
897 ;SET SINGLE OUPUT PIN
898 BANK0
899 BCF PORTB,1
900
901 RETURN
902
903 LABEL_RECEIVE_@_RF
904
905 ;BEGIN OF FOR-LOOP
906 ;SET VARIABLE U8 WITH CONSTANT
907 MOVLW 0X8
908 BANK0
909 MOVWF BYTE_COUNTER
910 INCF BYTE_COUNTER,F
911 LABEL_1018
912 BANK0
913 DECF BYTE_COUNTER,F
914 BTFSC STATUS,Z
915 GOTO LABEL_1019
916
917 ;BEGIN OF FOR-LOOP
918 ;SET VARIABLE U8 WITH CONSTANT
919 MOVLW 0X8
920 BANK0
921 MOVWF BIT_COUNTER
922 INCF BIT_COUNTER,F
923 LABEL_1020
924 BANK0
925 DECF BIT_COUNTER,F
926 BTFSC STATUS,Z
927 GOTO LABEL_1021
928
929 CALL LABEL_RECEIVE_RF_DATA_BIT
930
931 GOTO LABEL_1020
932 LABEL_1021
933 ;END OF FOR/WHILE-LOOP
934
935 ;SET INDIRECT VARIABLE WITH VARIABLE U8
936 ;SET U16 POINTER
937 BCF STATUS,IRP
938 BANK0
939 BTFSC POINTER_RF_IN_H,0
940 BSF STATUS,IRP
941 BANK0
942 MOVF POINTER_RF_IN,W
943 MOVWF FSR
944 MOVF RF_IN_DATA,W
945 MOVWF INDF
946
947 INCF POINTER_RF_IN,F
948 BTFSC STATUS,Z
949 INCF POINTER_RF_IN_H,F
950
951 GOTO LABEL_1018
952 LABEL_1019
953 ;END OF FOR/WHILE-LOOP
954
955 RETURN
956
957 LABEL_INIT
958
959 ;SET VARIABLE U16 WITH CONSTANT
960 MOVLW 0XA0
961 BANK0
962 MOVWF POINTER_RF_OUT
963 MOVLW 0X0
964 MOVWF POINTER_RF_OUT_H
965
966 ;SET VARIABLE U16 WITH CONSTANT
967 MOVLW 0X20
968 MOVWF POINTER_RF_IN
969 MOVLW 0X1
970 MOVWF POINTER_RF_IN_H
971
972 CLRF RF_PACKET_COUNTER
973
974 ;SET VARIABLE U8 WITH CONSTANT
975 MOVLW 0X8
976 MOVWF RF_FRAME_LENGTH
977
978 CLRF RF_FRAME_INDEX
979
980 ;CONFIGURE SINGLE PIN
981 BANK1
982 BSF TRISB,0
983
984 ;CONFIGURE SINGLE PIN
985 BSF TRISB,1
986
987 ;CONFIGURE SINGLE PIN
988 BSF TRISB,3
989
990 RETURN
991
992
993
994 END

```

11 PICLab Program (DISPLAY_1.0) for IC2

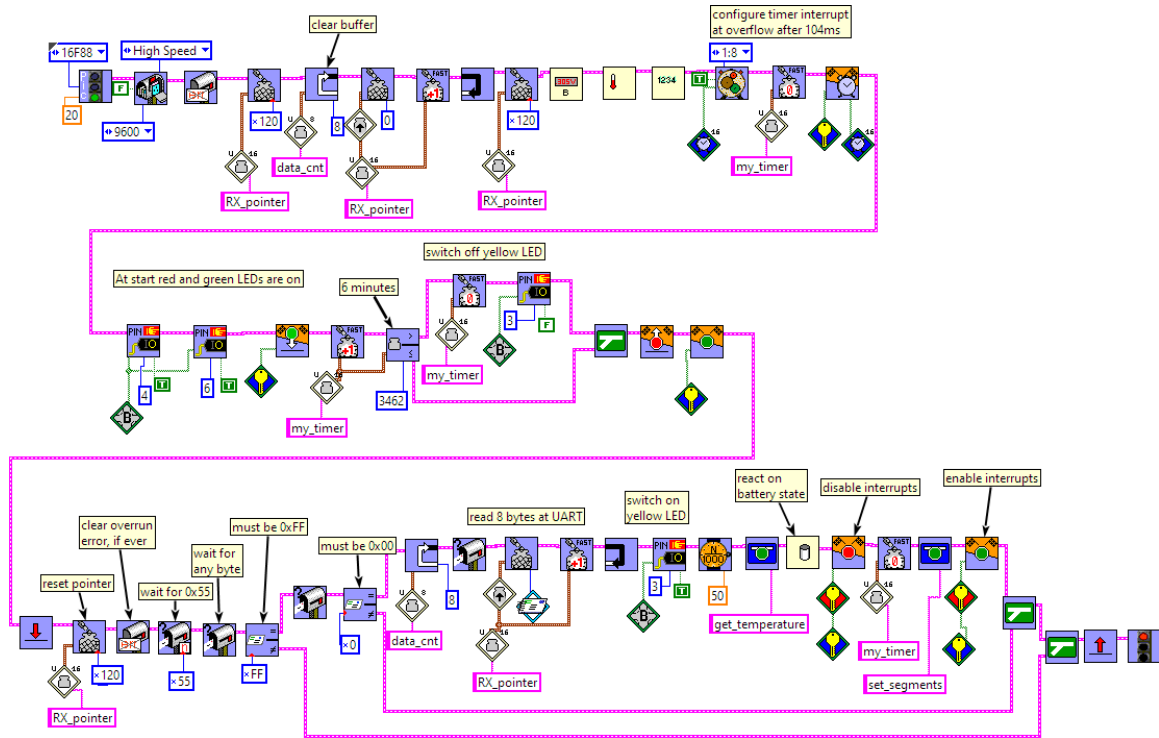


Figure 19: PICLAB main display program for IC2 (filename: *test_display5.vi*).

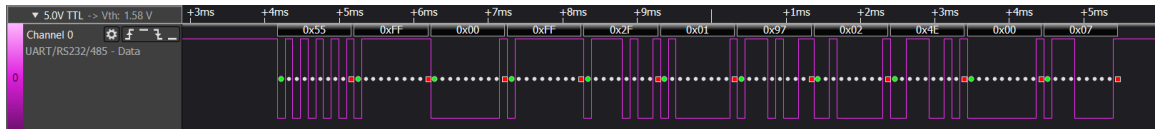


Figure 20: Packet (with 0x55 FF 00 header) received via UART. Byte 1 after header = TRX433 address; Byte 2 = running counter; Bytes 3&4 = Battery level (10-bit raw); Bytes 5&6 = Temperature (10-bit raw); Bytes 7&8 = Jumper voltage (10-bit raw)

12 Assembler code (DISPLAY_1.0) for IC2

```

1  ;-----
2  ;               Assembler code for test_display5.vi created with PICLab
3  ;-----
4
5      LIST      p=16F88
6
7      #include "P16F88.INC" ; Include header file
8
9      __CONFIG _CONFIG1, _BODEN_OFF&_CP_OFF&_PWRTE_ON&_WDT_OFF&_LVP_OFF&_MCLR_ON&_HS_OSC
10
11 ;*****Variable definitions*****
12
13 TEMPPORT8 EQU 0X20
14 TEMPX8 EQU 0X21
15 TEMPY8 EQU 0X22
16 RESULT8 EQU 0X23
17 TEMPX16 EQU 0X24
18 TEMPX16_H EQU 0X25
19 TEMPY16 EQU 0X26
20 TEMPY16_H EQU 0X27
21 RESULT16 EQU 0X28
22 RESULT16_H EQU 0X29
23 IDX16 EQU 0X2A
24 IDX16_H EQU 0X2B
25 TEMPYY EQU 0X2C
26 OP_SIGN8 EQU 0X2D
27 ADHBYTE EQU 0X2E
28 ADLBYTE EQU 0X2F
29 AD_RESULT EQU 0X30
30 AD_RESULT_H EQU 0X31

```

```

31 | ISR_TEMPSPORT8 EQU 0X32
32 | ISR_TEMPX8 EQU 0X33
33 | ISR_TEMPY8 EQU 0X34
34 | ISR_RESULTS8 EQU 0X35
35 | ISR_TEMPX16 EQU 0X36
36 | ISR_TEMPX16_H EQU 0X37
37 | ISR_TEMPY16 EQU 0X38
38 | ISR_TEMPY16_H EQU 0X39
39 | ISR_RESULT16 EQU 0X3A
40 | ISR_RESULT16_H EQU 0X3B
41 | ISR_IDX16 EQU 0X3C
42 | ISR_IDX16_H EQU 0X3D
43 | ISR_TEMPYY EQU 0X3E
44 | ISR_OP_SIGN8 EQU 0X3F
45 | ISR_STATUS EQU 0X40
46 | ISR_W EQU 0X41
47 | ISR_FSR EQU 0X42
48 | DATA_CNT EQU 0X43
49 | MY_TIMER EQU 0X44
50 | MY_TIMER_H EQU 0X45
51 | RX_POINTER EQU 0X46
52 | RX_POINTER_H EQU 0X47
53 | T_BCD_POINTER EQU 0X48
54 | T_BCD_POINTER_H EQU 0X49
55 | SEGMENT_INDEX EQU 0X4A
56 | CURRENT_DIGIT_CODE EQU 0X4B
57 | DIGIT_BLUE_CODE EQU 0X4C
58 | DISPLAY_STATE EQU 0X4D
59 | DIGIT_INDEX EQU 0X4E
60 | LOOKUP_TEMP EQU 0X4F
61 | DATA_POINTER EQU 0X50
62 | DATA_POINTER_H EQU 0X51
63 | AA1 EQU 0X52
64 | AA1_H EQU 0X53
65 | RAW EQU 0X54
66 | RAW_H EQU 0X55
67 | AA2 EQU 0X56
68 | AA2_H EQU 0X57
69 | BB1 EQU 0X58
70 | BB1_H EQU 0X59
71 | WARM EQU 0X5A
72 | WARM_H EQU 0X5B
73 | S_DATA_POINTER EQU 0X5C
74 | S_DATA_POINTER_H EQU 0X5D
75 | TMPX16 EQU 0X5E
76 | TMPX16_H EQU 0X5F
77 | DEC EQU 0X60
78 | NUMBER EQU 0X61
79 | NUMBER_H EQU 0X62
80 | J_DEC EQU 0X63
81 | DAT EQU 0X64
82 | DEC_PLACE EQU 0X65
83 | DEC_PLACE_H EQU 0X66
84 | VOLTAGE EQU 0X67
85 | VOLTAGE_H EQU 0X68
86 |
87 | ;*****Makro definitions and definitions of used operations*****
88 |
89 | BANK0 MACRO
90 |     BCF STATUS,RP0
91 |     BCF STATUS,RP1
92 |     ENDM
93 | BANK1 MACRO
94 |     BSF STATUS,RP0
95 |     BCF STATUS,RP1
96 |     ENDM
97 | BANK2 MACRO
98 |     BCF STATUS,RP0
99 |     BSF STATUS,RP1
100 |    ENDM
101 | BANK3 MACRO
102 |     BSF STATUS,RP0
103 |     BSF STATUS,RP1
104 |     ENDM
105 |
106 |     GOTO START
107 |     ORG 0X4
108 |     GOTO LABEL_ISR
109 |
110 | MULV16
111 |     CLRF RESULT16
112 |     CLRF RESULT16_H
113 | MULV16LOOP
114 |     BTFSC TEMPY16,0
115 |     CALL ADD16
116 |     BCF STATUS,C
117 |     RRF TEMPY16_H,F
118 |     RRF TEMPY16,F
119 |     BCF STATUS,C
120 |     RLF TEMPX16,F
121 |     RLF TEMPX16_H,F
122 |     MOVF TEMPY16,F
123 |     BTFSS STATUS,Z
124 |     GOTO MULV16LOOP
125 |     MOVF TEMPY16_H,F
126 |     BTFSS STATUS,Z
127 |     GOTO MULV16LOOP
128 |     RETURN
129 |
130 | ADD16
131 |     MOVF TEMPX16,W
132 |     ADDWF RESULT16
133 |     BTFSC STATUS,C
134 |     INCF RESULT16_H
135 |     MOVF TEMPX16_H,W
136 |     ADDWF RESULT16_H
137 |     RETURN
138 |
139 | SUMV16
140 |     MOVF TEMPY16,W
141 |     ADDWF TEMPX16
142 |     BTFSC STATUS,C
143 |     INCF TEMPX16_H
144 |     MOVF TEMPY16_H,W
145 |     ADDWF TEMPX16_H
146 |     RETURN
147 |
148 | DIV_I16
149 |     CLRF OP_SIGN8

```

```

150| BTFSS  TEMPX16_H,7 ;SKIP IF NEGATIVE
151| GOTO  CONTINUE_DIV_I16_1
152| COMF  TEMPX16,F ;CHANGE SIGN
153| COMF  TEMPX16_H,F
154| MOVLW 1
155| ADDWF TEMPX16
156| BTFSC STATUS,C
157| INCF  TEMPX16_H,F
158| COMF  OP_SIGN8,F
159| CONTINUE_DIV_I16_1
160| BTFSS  TEMPY16,7 ;SKIP IF NEGATIVE
161| GOTO  CONTINUE_DIV_I16_2
162| COMF  TEMPY16,F ;CHANGE SIGN
163| COMF  TEMPY16_H,F
164| MOVLW 1
165| ADDWF TEMPY16
166| BTFSC STATUS,C
167| INCF  TEMPY16_H,F
168| COMF  OP_SIGN8,F
169| CONTINUE_DIV_I16_2
170| CALL  DIVV16
171| BTFSS  OP_SIGN8,0 ;SKIP IF NEGATIVE
172| RETURN
173| COMF  RESULT16,F
174| COMF  RESULT16_H,F
175| MOVLW 1
176| ADDWF RESULT16
177| BTFSC STATUS,C
178| INCF  RESULT16_H,F
179| RETURN
180|
181| DIVV16
182| MOVF  TEMPY16,F
183| BTFSS  STATUS,Z
184| GOTO  ZERO_TEST_SKIPPED
185| MOVF  TEMPY16_H,F
186| BTFSC  STATUS,Z
187| RETURN
188| ZERO_TEST_SKIPPED
189| MOVLW 1
190| MOVWF  IDX16
191| CLRF  IDX16_H
192| CLRF  RESULT16
193| CLRF  RESULT16_H
194| SHIFT_IT16
195| BCF  STATUS,C
196| RLF  IDX16,F
197| RLF  IDX16_H,F
198| BCF  STATUS,C
199| RLF  TEMPY16,F
200| RLF  TEMPY16_H,F
201| BTFSS  TEMPY16_H,7
202| GOTO  SHIFT_IT16
203| DIVU16LOOP
204| CALL  SUB16
205| BTFSC  STATUS,C
206| GOTO  COUNTX
207| CALL  ADD16BIS
208| GOTO  FINALX
209| COUNTX
210| MOVF  IDX16,W
211| ADDWF RESULT16
212| BTFSC  STATUS,C
213| INCF  RESULT16_H,F
214| MOVF  IDX16_H,W
215| ADDWF RESULT16_H
216| FINALX
217| BCF  STATUS,C
218| RRF  TEMPY16_H,F
219| RRF  TEMPY16,F
220| BCF  STATUS,C
221| RRF  IDX16_H,F
222| RRF  IDX16,F
223| BTFSS  STATUS,C
224| GOTO  DIVU16LOOP
225| RETURN
226|
227| SUB16
228| MOVF  TEMPY16_H,W
229| MOVWF TEMPYY
230| MOVF  TEMPY16,W
231| SUBWF TEMPX16
232| BTFSS  STATUS,C
233| INCF  TEMPYY,F
234| MOVF  TEMPYY,W
235| SUBWF TEMPX16_H
236| RETURN
237|
238| ADD16BIS
239| MOVF  TEMPY16,W
240| ADDWF TEMPX16
241| BTFSC  STATUS,C
242| INCF  TEMPX16_H,F
243| MOVF  TEMPY16_H,W
244| ADDWF TEMPX16_H
245| RETURN
246|
247| SUBY16
248| MOVF  TEMPY16,W
249| SUBWF TEMPX16
250| BTFSS  STATUS,C
251| INCF  TEMPY16_H,F
252| MOVF  TEMPY16_H,W
253| SUBWF TEMPX16_H
254| RETURN
255|
256| ABSV16
257| BTFSS  TEMPY16_H,7
258| RETURN
259| COMF  TEMPY16,F
260| COMF  TEMPY16_H,F
261| MOVLW 1
262| ADDWF TEMPY16
263| BTFSC  STATUS,C
264| INCF  TEMPY16_H,F
265| BCF  TEMPY16_H,7 ;CLEAR SIGN FLAG
266| RETURN
267|
268| ORV8

```

```

269 MOVF TEMPY8,W
270 IORWF TEMPX8
271 RETURN
272
273 GRU8U8
274 MOVF TEMPX8,W
275 SUBWF TEMPY8
276 BTFSS STATUS,C
277 RETLW 0
278 RETLW 1
279
280 GRU16U16
281 MOVF TEMPX16,W
282 SUBWF TEMPY16
283 MOVF TEMPX16_H,W
284 BTFSS STATUS,C
285 INCFSSZ TEMPX16_H,W
286 SUBWF TEMPY16_H
287 BTFSS STATUS,C
288 RETLW 0
289 RETLW 1
290
291 GRI16I16
292 CLRF IDX16
293 BSF IDX16,0
294 BTFSS TEMPX16_H,7
295 BCF IDX16,0
296 CLRF RESULT16
297 BSF RESULT16,0
298 BTFSS TEMPY16_H,7
299 BCF RESULT16,0
300 MOVF RESULT16,W
301 SUBWF IDX16
302 BTFSS IDX16,0
303 GOTO GRU16U16
304 BTFSS STATUS,C
305 RETLW 0
306 RETLW 1
307
308 EQU16U16
309 MOVF TEMPX16,W
310 SUBWF TEMPY16
311 BTFSS STATUS,Z
312 RETLW 0
313 MOVF TEMPX16_H,W
314 SUBWF TEMPY16_H
315 BTFSS STATUS,Z
316 RETLW 0
317 RETLW 1
318
319 EQU8U8
320 MOVF TEMPX8,W
321 SUBWF TEMPY8
322 BTFSS STATUS,Z ;SKIP IF ZERO ->IDENTICAL
323 RETLW 0
324 RETLW 1
325
326 ISR_SHLV8
327 BCF STATUS,C
328 RLF ISR_TEMPY8,F
329 RETURN
330
331 ISR_EQU8U8
332 MOVF ISR_TEMPX8,W
333 SUBWF ISR_TEMPY8
334 BTFSS STATUS,Z ;SKIP IF ZERO ->IDENTICAL
335 RETLW 0
336 RETLW 1
337
338 ISR_GRU8U8
339 MOVF ISR_TEMPX8,W
340 SUBWF ISR_TEMPY8
341 BTFSS STATUS,C
342 RETLW 0
343 RETLW 1
344
345 ISR_GRU16U16
346 MOVF ISR_TEMPX16,W
347 SUBWF ISR_TEMPY16
348 MOVF ISR_TEMPX16_H,W
349 BTFSS STATUS,C
350 INCFSSZ ISR_TEMPX16_H,W
351 SUBWF ISR_TEMPY16_H
352 BTFSS STATUS,C
353 RETLW 0
354 RETLW 1
355
356 ;*****BEGIN OF MAIN PROGRAM*****
357
358 START
359
360 ;INITIALIZE PORT A
361 BANK0
362 CLRF PORTA
363 BANK1
364 CLRF ANSEL
365 MOVLW 0X07
366 MOVWF CMCON
367 MOVLW 0X0
368 MOVWF TRISA
369
370 ;INITIALIZE PORT B
371 BANK0
372 CLRF PORTB
373 BANK1
374 MOVLW 0X0
375 MOVWF TRISB
376
377 ;CONFIGURE UART
378 BSF TRISB,2
379 BSF TRISB,5
380 MOVLW 0X81
381 MOVWF SPBRG
382 BSF TXSTA,BRGH
383 BCF TXSTA,SYNC
384 BANK0
385 BSF RCSTA,SPEN
386 BCF RCSTA,RX9
387 BSF RCSTA,CREN

```

```

388 BANK1
389 BCF TXSTA, TX9
390 BSF TXSTA, TXEN
391
392 ;CLEAR MAIL ERROR
393 BANK0
394 BTFSS RCSTA, OERR
395 GOTO LABEL_1002
396 BCF RCSTA, CREN
397 BSF RCSTA, CREN
398 LABEL_1002
399
400 ;SET VARIABLE U16 WITH CONSTANT
401 MOVLW 0X20
402 BANK0
403 MOVWF RX_POINTER
404 MOVLW 0X1
405 MOVWF RX_POINTER_H
406
407 ;BEGIN OF FOR-LOOP
408 ;SET VARIABLE U8 WITH CONSTANT
409 MOVLW 0X8
410 MOVWF DATA_CNT
411 INCF DATA_CNT, F
412 LABEL_1003
413 BANK0
414 DECF DATA_CNT, F
415 BTFSC STATUS, Z
416 GOTO LABEL_1004
417
418 ;SET INDIRECT VARIABLE WITH CONSTANT
419 ;SET U16 POINTER
420 BCF STATUS, IRP
421 BANK0
422 BTFSC RX_POINTER_H, 0
423 BSF STATUS, IRP
424 BANK0
425 MOVF RX_POINTER, W
426 MOVWF FSR
427 MOVLW 0X0
428 MOVWF INDF
429
430 INCF RX_POINTER, F
431 BTFSC STATUS, Z
432 INCF RX_POINTER_H, F
433
434 GOTO LABEL_1003
435 LABEL_1004
436 ;END OF FOR/WHILE-LOOP
437
438 ;SET VARIABLE U16 WITH CONSTANT
439 MOVLW 0X20
440 BANK0
441 MOVWF RX_POINTER
442 MOVLW 0X1
443 MOVWF RX_POINTER_H
444
445 ;CONFIGURE TMR2
446 MOVLW B'100100'
447 MOVWF T2CON
448 MOVLW 0XFA
449 BANK1
450 MOVWF PR2
451
452 ;CONFIGURE VIRTUAL PORT
453 ;CONFIGURE PORT OUTPUTS
454 MOVF TRISA, W
455 ANDLW 0XFO
456 MOVWF TRISA
457
458 ;SET VIRTUAL PORT VALUE WITH CONSTANT
459 MOVLW 0XF
460 BANK0
461 MOVWF TEMPPORT8
462 CALL OUT_GENERIC_PORT1
463
464 ;SET SINGLE OUPUT PIN
465 BANK0
466 BSF PORTB, 0
467
468 ;SET SINGLE OUPUT PIN
469 BSF PORTB, 1
470
471 ;WAIT
472 MOVLW 0XAA
473 MOVWF TEMPYY
474 LABEL_1009
475 MOVLW 0X1D
476 MOVWF IDX16
477 LABEL_1010
478 MOVLW 0XA7
479 MOVWF IDX16_H
480 LABEL_1011
481 DECFSZ IDX16_H, F
482 GOTO LABEL_1011
483 DECFSZ IDX16, F
484 GOTO LABEL_1010
485 DECFSZ TEMPYY, F
486 GOTO LABEL_1009
487 NOP
488 NOP
489 NOP
490 NOP
491 NOP
492 NOP
493 NOP
494
495 BANK0
496 CLRF DIGIT_INDEX
497
498 CLRF SEGMENT_INDEX
499
500 ;SET VARIABLE U16 WITH CONSTANT
501 MOVLW 0X10
502 MOVWF T_BCD_POINTER
503 MOVLW 0X1
504 MOVWF T_BCD_POINTER_H
505
506 ;SET INDIRECT VARIABLE WITH CONSTANT

```

```

507 ;SET U16 POINTER
508 BCF STATUS,IRP
509 BTFSK T_BCD_POINTER_H,0
510 BSF STATUS,IRP
511 BANKO
512 MOVF T_BCD_POINTER,W
513 MOVWF FSR
514 MOVLW 0X4F
515 MOVWF INDF
516
517 INCF T_BCD_POINTER,F
518 BTFSK STATUS,Z
519 INCF T_BCD_POINTER_H,F
520
521 ;SET INDIRECT VARIABLE WITH CONSTANT
522 ;SET U16 POINTER
523 BCF STATUS,IRP
524 BANKO
525 BTFSK T_BCD_POINTER_H,0
526 BSF STATUS,IRP
527 BANKO
528 MOVF T_BCD_POINTER,W
529 MOVWF FSR
530 MOVLW 0X5B
531 MOVWF INDF
532
533 INCF T_BCD_POINTER,F
534 BTFSK STATUS,Z
535 INCF T_BCD_POINTER_H,F
536
537 ;SET INDIRECT VARIABLE WITH CONSTANT
538 ;SET U16 POINTER
539 BCF STATUS,IRP
540 BANKO
541 BTFSK T_BCD_POINTER_H,0
542 BSF STATUS,IRP
543 BANKO
544 MOVF T_BCD_POINTER,W
545 MOVWF FSR
546 MOVLW 0X6
547 MOVWF INDF
548
549 INCF T_BCD_POINTER,F
550 BTFSK STATUS,Z
551 INCF T_BCD_POINTER_H,F
552
553 ;SET INDIRECT VARIABLE WITH CONSTANT
554 ;SET U16 POINTER
555 BCF STATUS,IRP
556 BANKO
557 BTFSK T_BCD_POINTER_H,0
558 BSF STATUS,IRP
559 BANKO
560 MOVF T_BCD_POINTER,W
561 MOVWF FSR
562 MOVLW 0X40
563 MOVWF INDF
564
565 ;SET VARIABLE U16 WITH CONSTANT
566 MOVLW 0X10
567 MOVWF T_BCD_POINTER
568 MOVLW 0X1
569 MOVWF T_BCD_POINTER_H
570
571 ;CALL LOOKUP-TABLE
572 ;SET VARIABLE US WITH VARIABLE US
573 BANKO
574 MOVF DIGIT_INDEX,W
575 MOVWF LOOKUP_TEMP
576 CALL LABEL_LOOKUP_DIGIT_SELECT
577 BANKO
578 MOVWF DIGIT_BLUE_CODE
579
580 ;SET VIRTUAL PORT VALUE WITH VARIABLE US
581 MOVF DIGIT_BLUE_CODE,W
582 MOVWF TEMPPORT8
583 CALL OUT_GENERIC_PORT1
584
585 ;SET VARIABLE US WITH INDIRECT VARIABLE
586 ;SET U16 POINTER
587 BCF STATUS,IRP
588 BANKO
589 BTFSK T_BCD_POINTER_H,0
590 BSF STATUS,IRP
591 BANKO
592 MOVF T_BCD_POINTER,W
593 MOVWF FSR
594 MOVF INDF,W
595 MOVWF CURRENT_DIGIT_CODE
596
597 ;SET VARIABLE US WITH CONSTANT
598 MOVLW 0X0
599 MOVWF DISPLAY_STATE
600
601 ;CLEAR TIMER
602 CLRF TMR2
603
604 ;START MONITORING INTERRUPTS
605 BCF PIR1,TMR2IF
606 BANK1
607 BSF PIE1,TMR2IE
608 BSF INTCON,GIE
609 BSF INTCON,PEIE
610
611 ;CONFIGURE TMR1
612 MOVLW B'110001'
613 BANKO
614 MOVWF T1CON
615
616 CLRF MY_TIMER
617 CLRF MY_TIMER_H
618
619 ;SET SINGLE OUPUT PIN
620 BSF PORTB,4
621
622 ;SET SINGLE OUPUT PIN
623 BSF PORTB,6
624
625 ;START MONITORING INTERRUPTS

```

```

626 BCF PIR1,TMR1IF
627 BANK1
628 BSF PIE1,TMR1IE
629 BSF INTCON,GIE
630 BSF INTCON,PEIE
631
632 LABEL_0
633
634 ;SET VARIABLE U16 WITH CONSTANT
635 MOVLW 0X20
636 BANK0
637 MOVWF RX_POINTER
638 MOVLW 0X1
639 MOVWF RX_POINTER_H
640
641 ;CLEAR MAIL ERROR
642 BTFFS RCSTA,0ERR
643 GOTO LABEL_1050
644 BCF RCSTA,CREN
645 BSF RCSTA,CREN
646 LABEL_1050
647
648 LABEL_1051
649
650 ;CLEAR MAIL ERROR
651 BANK0
652 BTFFS RCSTA,0ERR
653 GOTO LABEL_1053
654 BCF RCSTA,CREN
655 BSF RCSTA,CREN
656 LABEL_1053
657
658 LABEL_1052
659
660 ;CHECK IF OVERRUN-ERROR
661 BANK0
662 BTFFS RCSTA,1
663
664 GOTO LABEL_1051
665
666 ;WAIT FOR MAIL
667 BANK0
668 LABEL_1054
669 BTFFS PIR1,RCIF
670 GOTO LABEL_1054
671
672 ;BEGIN OF IF-STRUCTURE
673 ;COMPARE-OPERATION
674 ;SET VARIABLE U8 WITH SFR
675 BANK0
676 MOVF RCREG,W
677 MOVWF TEMPX8
678 ;SET VARIABLE U8 WITH CONSTANT
679 MOVLW 0X55
680 MOVWF TEMPY8
681 CALL EQU8U8
682 BANK0
683 MOVWF RESULTS
684 BTFFS RESULTS,0
685 GOTO LABEL_1055
686
687 GOTO LABEL_1052
688
689 GOTO LABEL_1056
690 LABEL_1055
691
692 LABEL_1056
693 ;END OF IF-STRUCTURE
694
695 ;WAIT FOR MAIL
696 BANK0
697 LABEL_1057
698 BTFFS PIR1,RCIF
699 GOTO LABEL_1057
700
701 ;BEGIN OF IF-STRUCTURE
702 ;COMPARE-OPERATION
703 ;SET VARIABLE U8 WITH SFR
704 BANK0
705 MOVF RCREG,W
706 MOVWF TEMPX8
707 ;SET VARIABLE U8 WITH CONSTANT
708 MOVLW 0XFF
709 MOVWF TEMPY8
710 CALL EQU8U8
711 BANK0
712 MOVWF RESULTS
713 BTFFS RESULTS,0
714 GOTO LABEL_1058
715
716 GOTO LABEL_1070
717 LABEL_1058
718
719 ;WAIT FOR MAIL
720 BANK0
721 LABEL_1059
722 BTFFS PIR1,RCIF
723 GOTO LABEL_1059
724
725 ;BEGIN OF IF-STRUCTURE
726 ;COMPARE-OPERATION
727 ;SET VARIABLE U8 WITH SFR
728 BANK0
729 MOVF RCREG,W
730 MOVWF TEMPX8
731 ;SET VARIABLE U8 WITH CONSTANT
732 MOVLW 0X0
733 MOVWF TEMPY8
734 CALL EQU8U8
735 BANK0
736 MOVWF RESULTS
737 BTFFS RESULTS,0
738 GOTO LABEL_1060
739
740 GOTO LABEL_1069
741 LABEL_1060
742
743 ;BEGIN OF FOR-LOOP
744 ;SET VARIABLE U8 WITH CONSTANT

```



```

745      MOVLW  0X8
746      BANK0
747      MOVWF  DATA_CNT
748      INCF  DATA_CNT,F
749      LABEL_1061
750      BANK0
751      DECF  DATA_CNT,F
752      BTFSC STATUS,Z
753      GOTO  LABEL_1062
754
755      ;WAIT FOR MAIL
756      BANK0
757      LABEL_1063
758      BTFSS PIR1,RCIF
759      GOTO  LABEL_1063
760
761      ;SET INDIRECT VARIABLE WITH SFR
762      ;SET U16 POINTER
763      BCF  STATUS,IRP
764      BANK0
765      BTFSC RX_POINTER_H,0
766      BSF  STATUS,IRP
767      BANK0
768      MOVF  RX_POINTER,W
769      MOVWF FSR
770      MOVF  RCREG,W
771      MOVWF INDF
772
773      INCF  RX_POINTER,F
774      BTFSC STATUS,Z
775      INCF  RX_POINTER_H,F
776
777      GOTO  LABEL_1061
778      LABEL_1062
779      ;END OF FOR/WHILE-LOOP
780
781      ;SET SINGLE OUPUT PIN
782      BANK0
783      BSF  PORTB,3
784
785      ;WAIT
786      MOVLW  0XA3
787      MOVWF TEMPYY
788      LABEL_1064
789      MOVLW  0X2
790      MOVWF IDX16
791      LABEL_1065
792      MOVLW  0XFB
793      MOVWF IDX16_H
794      LABEL_1066
795      DECFSZ IDX16_H,F
796      GOTO  LABEL_1066
797      DECFSZ IDX16,F
798      GOTO  LABEL_1065
799      DECFSZ TEMPYY,F
800      GOTO  LABEL_1064
801      NOP
802
803      CALL  LABEL_GET_TEMPERATURE
804
805      ;SET VARIABLE U16 WITH CONSTANT
806      MOVLW  0X22
807      BANK0
808      MOVWF DATA_POINTER
809      MOVLW  0X1
810      MOVWF DATA_POINTER_H
811
812      ;SET VARIABLE U16 WITH INDIRECT VARIABLE
813      ;SET U16 POINTER
814      BCF  STATUS,IRP
815      BTFSC DATA_POINTER_H,0
816      BSF  STATUS,IRP
817      BANK0
818      MOVF  DATA_POINTER,W
819      MOVWF FSR
820      MOVF  INDF,W
821      MOVWF VOLTAGE
822      CLRF  VOLTAGE_H
823
824      ;OPERATION
825      ;SET VARIABLE U16 WITH VARIABLE U16
826      MOVF  VOLTAGE,W
827      MOVWF TEMP16
828      MOVF  VOLTAGE_H,W
829      MOVWF TEMP16_H
830      ;SET VARIABLE U16 WITH CONSTANT
831      MOVLW  0X0
832      MOVWF TEMPY16
833      MOVLW  0X1
834      MOVWF TEMPY16_H
835      CALL  MULV16
836      ;SET VARIABLE U16 WITH VARIABLE U16
837      BANK0
838      MOVF  RESULT16,W
839      MOVWF VOLTAGE
840      MOVF  RESULT16_H,W
841      MOVWF VOLTAGE_H
842
843      ;SET VARIABLE U16 WITH CONSTANT
844      MOVLW  0X23
845      MOVWF DATA_POINTER
846      MOVLW  0X1
847      MOVWF DATA_POINTER_H
848
849      ;OPERATION
850      ;SET VARIABLE U16 WITH VARIABLE U16
851      MOVF  VOLTAGE,W
852      MOVWF TEMP16
853      MOVF  VOLTAGE_H,W
854      MOVWF TEMP16_H
855      ;SET VARIABLE U16 WITH INDIRECT VARIABLE
856      ;SET U16 POINTER
857      BCF  STATUS,IRP
858      BTFSC DATA_POINTER_H,0
859      BSF  STATUS,IRP
860      BANK0
861      MOVF  DATA_POINTER,W
862      MOVWF FSR
863      MOVF  INDF,W

```

```

864      MOVWF TEMPY16
865      CLRF TEMPY16_H
866      CALL SUMV16
867      ;SET VARIABLE U16 WITH VARIABLE U16
868      BANK0
869      MOVF  TEMPX16,W
870      MOVWF VOLTAGE
871      MOVF  TEMPX16_H,W
872      MOVWF VOLTAGE_H
873
874      ;BEGIN OF IF-STRUCTURE
875      ;COMPARE-OPERATION
876      ;SET VARIABLE U16 WITH VARIABLE U16
877      MOVF  VOLTAGE,W
878      MOVWF TEMPX16
879      MOVF  VOLTAGE_H,W
880      MOVWF TEMPX16_H
881      ;SET VARIABLE U16 WITH CONSTANT
882      MOVLW 0X88
883      MOVWF TEMPY16
884      MOVLW 0X1
885      MOVWF TEMPY16_H
886      CALL GRU16U16
887      BANK0
888      MOVWF RESULTS
889      BTFSK RESULTS,0
890      GOTO LABEL_1067
891
892      ;SET SINGLE OUPUT PIN
893      BANK0
894      BSF PORTB,4
895
896      ;SET SINGLE OUPUT PIN
897      BCF PORTB,6
898
899      GOTO LABEL_1068
900 LABEL_1067
901
902      ;SET SINGLE OUPUT PIN
903      BANK0
904      BCF PORTB,4
905
906      ;SET SINGLE OUPUT PIN
907      BSF PORTB,6
908
909 LABEL_1068
910 ;END OF IF-STRUCTURE
911
912 ;STOP MONITORING INTERRUPTS
913 BANK1
914 BCF PIE1,TMR2IE
915 BCF PIE1,TMR1IE
916
917
918 BANK0
919 CLRF MY_TIMER
920 CLRF MY_TIMER_H
921
922 CALL LABEL_SET_SEGMENTS
923
924 ;START MONITORING INTERRUPTS
925 BANK0
926 BCF PIR1,TMR2IF
927 BCF PIR1,TMR1IF
928 BANK1
929 BSF PIE1,TMR2IE
930 BSF PIE1,TMR1IE
931 BSF INTCON,GIE
932 BSF INTCON,PEIE
933
934 LABEL_1069
935 ;END OF IF-STRUCTURE
936
937 LABEL_1070
938 ;END OF IF-STRUCTURE
939
940 GOTO LABEL_0
941
942 LABEL_1071
943
944 GOTO LABEL_1071
945
946 ;*****SUBROUTINES*****
947
948 LABEL_GET_TEMPERATURE
949
950 ;SET VARIABLE U16 WITH CONSTANT
951 MOVLW 0X24
952 BANK0
953 MOVWF DATA_POINTER
954 MOVLW 0X1
955 MOVWF DATA_POINTER_H
956
957 ;SET VARIABLE U16 WITH INDIRECT VARIABLE
958 ;SET U16 POINTER
959 BCF STATUS,IRP
960 BTFSK DATA_POINTER_H,0
961 BSF STATUS,IRP
962 BANK0
963 MOVF DATA_POINTER,W
964 MOVWF FSR
965 MOVF INDF,W
966 MOVWF RAW
967 CLRF RAW_H
968
969 ;OPERATION
970 ;SET VARIABLE U16 WITH VARIABLE U16
971 MOVF RAW,W
972 MOVWF TEMPX16
973 MOVF RAW_H,W
974 MOVWF TEMPX16_H
975 ;SET VARIABLE U16 WITH CONSTANT
976 MOVLW 0X0
977 MOVWF TEMPY16
978 MOVLW 0X1
979 MOVWF TEMPY16_H
980 CALL MULV16
981 ;SET VARIABLE U16 WITH VARIABLE U16
982 BANK0

```

```

983 MOVF RESULT16,W
984 MOVWF RAW
985 MOVF RESULT16_H,W
986 MOVWF RAW_H
987
988 ;SET VARIABLE U16 WITH CONSTANT
989 MOVLW 0X25
990 MOVWF DATA_POINTER
991 MOVLW 0X1
992 MOVWF DATA_POINTER_H
993
994 ;OPERATION
995 ;SET VARIABLE U16 WITH VARIABLE U16
996 MOVF RAW,W
997 MOVWF TEMPX16
998 MOVF RAW_H,W
999 MOVWF TEMPX16_H
1000 ;SET VARIABLE U16 WITH INDIRECT VARIABLE
1001 ;SET U16 POINTER
1002 BCF STATUS,IRP
1003 BTFSK DATA_POINTER_H,0
1004 BSF STATUS,IRP
1005 BANK0
1006 MOVF DATA_POINTER,W
1007 MOVWF FSR
1008 MOVF INDF,W
1009 MOVWF TEMPY16
1010 CLRWF TEMPY16_H
1011 CALL SUMV16
1012 ;SET VARIABLE U16 WITH VARIABLE U16
1013 BANK0
1014 MOVF TEMPX16,W
1015 MOVWF RAW
1016 MOVF TEMPX16_H,W
1017 MOVWF RAW_H
1018
1019 ;BEGIN OF IF-STRUCTURE
1020 ;COMPARE-OPERATION
1021 ;SET VARIABLE U16 WITH VARIABLE U16
1022 MOVF RAW,W
1023 MOVWF TEMPX16
1024 MOVF RAW_H,W
1025 MOVWF TEMPX16_H
1026 ;SET VARIABLE U16 WITH CONSTANT
1027 MOVLW 0XC8
1028 MOVWF TEMPY16
1029 MOVLW 0X0
1030 MOVWF TEMPY16_H
1031 CALL GRU16U16
1032 BANK0
1033 MOVWF RESULTS
1034 BTFSK RESULTS,0
1035 GOTO LABEL_1028
1036
1037 ;BEGIN OF IF-STRUCTURE
1038 ;COMPARE-OPERATION
1039 ;SET VARIABLE U16 WITH VARIABLE U16
1040 BANK0
1041 MOVF RAW,W
1042 MOVWF TEMPX16
1043 MOVF RAW_H,W
1044 MOVWF TEMPX16_H
1045 ;SET VARIABLE U16 WITH CONSTANT
1046 MOVLW 0X2C
1047 MOVWF TEMPY16
1048 MOVLW 0X1
1049 MOVWF TEMPY16_H
1050 CALL GRU16U16
1051 BANK0
1052 MOVWF RESULTS
1053 BTFSK RESULTS,0
1054 GOTO LABEL_1029
1055
1056 ;BEGIN OF IF-STRUCTURE
1057 ;COMPARE-OPERATION
1058 ;SET VARIABLE U16 WITH VARIABLE U16
1059 BANK0
1060 MOVF RAW,W
1061 MOVWF TEMPX16
1062 MOVF RAW_H,W
1063 MOVWF TEMPX16_H
1064 ;SET VARIABLE U16 WITH CONSTANT
1065 MOVLW 0XF4
1066 MOVWF TEMPY16
1067 MOVLW 0X1
1068 MOVWF TEMPY16_H
1069 CALL GRU16U16
1070 BANK0
1071 MOVWF RESULTS
1072 BTFSK RESULTS,0
1073 GOTO LABEL_1030
1074
1075 ;BEGIN OF IF-STRUCTURE
1076 ;COMPARE-OPERATION
1077 ;SET VARIABLE U16 WITH VARIABLE U16
1078 BANK0
1079 MOVF RAW,W
1080 MOVWF TEMPX16
1081 MOVF RAW_H,W
1082 MOVWF TEMPX16_H
1083 ;SET VARIABLE U16 WITH CONSTANT
1084 MOVLW 0XBC
1085 MOVWF TEMPY16
1086 MOVLW 0X2
1087 MOVWF TEMPY16_H
1088 CALL GRU16U16
1089 BANK0
1090 MOVWF RESULTS
1091 BTFSK RESULTS,0
1092 GOTO LABEL_1031
1093
1094 ;BEGIN OF IF-STRUCTURE
1095 ;COMPARE-OPERATION
1096 ;SET VARIABLE U16 WITH VARIABLE U16
1097 BANK0
1098 MOVF RAW,W
1099 MOVWF TEMPX16
1100 MOVF RAW_H,W
1101 MOVWF TEMPX16_H

```

```

1102 | ;SET VARIABLE I16 WITH CONSTANT
1103 | MOVLW 0X84
1104 | MOVWF TEMPY16
1105 | MOVLW 0X3
1106 | MOVWF TEMPY16_H
1107 | CALL GRU16U16
1108 | BANKO
1109 | MOVWF RESULTS
1110 | BTFSK RESULTS,0
1111 | GOTO LABEL_1032
1112 |
1113 | ;SET VARIABLE I16 WITH CONSTANT
1114 | MOVLW 0X1E
1115 | BANKO
1116 | MOVWF AA1
1117 | MOVLW 0X0
1118 | MOVWF AA1_H
1119 |
1120 | ;SET VARIABLE I16 WITH CONSTANT
1121 | MOVLW 0XD
1122 | MOVWF AA2
1123 | MOVLW 0X0
1124 | MOVWF AA2_H
1125 |
1126 | ;SET VARIABLE I16 WITH CONSTANT
1127 | MOVLW 0X4E
1128 | MOVWF BB1
1129 | MOVLW 0X7
1130 | MOVWF BB1_H
1131 |
1132 | GOTO LABEL_1033
1133 | LABEL_1032
1134 |
1135 | ;SET VARIABLE I16 WITH CONSTANT
1136 | MOVLW 0X7
1137 | BANKO
1138 | MOVWF AA1
1139 | MOVLW 0X0
1140 | MOVWF AA1_H
1141 |
1142 | ;SET VARIABLE I16 WITH CONSTANT
1143 | MOVLW 0X5
1144 | MOVWF AA2
1145 | MOVLW 0X0
1146 | MOVWF AA2_H
1147 |
1148 | ;SET VARIABLE I16 WITH CONSTANT
1149 | MOVLW 0X1A
1150 | MOVWF BB1
1151 | MOVLW 0X4
1152 | MOVWF BB1_H
1153 |
1154 | LABEL_1033
1155 | ;END OF IF-STRUCTURE
1156 |
1157 | GOTO LABEL_1034
1158 | LABEL_1031
1159 |
1160 | ;SET VARIABLE I16 WITH CONSTANT
1161 | MOVLW 0X2A
1162 | BANKO
1163 | MOVWF AA1
1164 | MOVLW 0X0
1165 | MOVWF AA1_H
1166 |
1167 | ;SET VARIABLE I16 WITH CONSTANT
1168 | MOVLW 0X32
1169 | MOVWF AA2
1170 | MOVLW 0X0
1171 | MOVWF AA2_H
1172 |
1173 | ;SET VARIABLE I16 WITH CONSTANT
1174 | MOVLW 0XA3
1175 | MOVWF BB1
1176 | MOVLW 0X2
1177 | MOVWF BB1_H
1178 |
1179 | LABEL_1034
1180 | ;END OF IF-STRUCTURE
1181 |
1182 | GOTO LABEL_1035
1183 | LABEL_1030
1184 |
1185 | ;SET VARIABLE I16 WITH CONSTANT
1186 | MOVLW 0X3C
1187 | BANKO
1188 | MOVWF AA1
1189 | MOVLW 0X0
1190 | MOVWF AA1_H
1191 |
1192 | ;SET VARIABLE I16 WITH CONSTANT
1193 | MOVLW 0X3B
1194 | MOVWF AA2
1195 | MOVLW 0X0
1196 | MOVWF AA2_H
1197 |
1198 | ;SET VARIABLE I16 WITH CONSTANT
1199 | MOVLW 0XF8
1200 | MOVWF BB1
1201 | MOVLW 0X2
1202 | MOVWF BB1_H
1203 |
1204 | LABEL_1035
1205 | ;END OF IF-STRUCTURE
1206 |
1207 | GOTO LABEL_1036
1208 | LABEL_1029
1209 |
1210 | ;SET VARIABLE I16 WITH CONSTANT
1211 | MOVLW 0I64
1212 | BANKO
1213 | MOVWF AA1
1214 | MOVLW 0X0
1215 | MOVWF AA1_H
1216 |
1217 | ;SET VARIABLE I16 WITH CONSTANT
1218 | MOVLW 0I3E
1219 | MOVWF AA2
1220 | MOVLW 0X0

```

```

1221                 MOVWF AA2_H
1222
1223                 ;SET VARIABLE I16 WITH CONSTANT
1224                 MOVLW 0XAC
1225                 MOVWF BB1
1226                 MOVLW 0X3
1227                 MOVWF BB1_H
1228
1229                 LABEL_1036
1230                 ;END OF IF-STRUCTURE
1231
1232                 GOTO LABEL_1037
1233 LABEL_1028
1234
1235                 ;SET VARIABLE I16 WITH CONSTANT
1236                 MOVLW 0X67
1237                 BANKO
1238                 MOVWF AA1
1239                 MOVLW 0X0
1240                 MOVWF AA1_H
1241
1242                 ;SET VARIABLE I16 WITH CONSTANT
1243                 MOVLW 0X2E
1244                 MOVWF AA2
1245                 MOVLW 0X0
1246                 MOVWF AA2_H
1247
1248                 ;SET VARIABLE I16 WITH CONSTANT
1249                 MOVLW 0X24
1250                 MOVWF BB1
1251                 MOVLW 0X4
1252                 MOVWF BB1_H
1253
1254 LABEL_1037
1255 ;END OF IF-STRUCTURE
1256
1257 ;SET VARIABLE I16 WITH VARIABLE U16
1258 BANKO
1259 MOVF RAW,W
1260 MOVWF WARM
1261 MOVF RAW_H,W
1262 MOVWF WARM_H
1263
1264 ;OPERATION
1265 ;SET VARIABLE U16 WITH VARIABLE I16
1266 MOVF WARM,W
1267 MOVWF TEMPX16
1268 MOVF WARM_H,W
1269 MOVWF TEMPX16_H
1270 ;SET VARIABLE U16 WITH VARIABLE I16
1271 MOVF AA1,W
1272 MOVWF TEMPY16
1273 MOVF AA1_H,W
1274 MOVWF TEMPY16_H
1275 CALL MUL16
1276 ;SET VARIABLE I16 WITH VARIABLE U16
1277 BANKO
1278 MOVF RESULT16,W
1279 MOVWF WARM
1280 MOVF RESULT16_H,W
1281 MOVWF WARM_H
1282
1283 ;OPERATION
1284 ;SET VARIABLE U16 WITH VARIABLE I16
1285 MOVF WARM,W
1286 MOVWF TEMPX16
1287 MOVF WARM_H,W
1288 MOVWF TEMPX16_H
1289 ;SET VARIABLE U16 WITH VARIABLE I16
1290 MOVF AA2,W
1291 MOVWF TEMPY16
1292 MOVF AA2_H,W
1293 MOVWF TEMPY16_H
1294 CALL DIV_I16
1295 ;SET VARIABLE I16 WITH VARIABLE U16
1296 BANKO
1297 MOVF RESULT16,W
1298 MOVWF WARM
1299 MOVF RESULT16_H,W
1300 MOVWF WARM_H
1301
1302 ;OPERATION
1303 ;SET VARIABLE U16 WITH VARIABLE I16
1304 MOVF WARM,W
1305 MOVWF TEMPX16
1306 MOVF WARM_H,W
1307 MOVWF TEMPX16_H
1308 ;SET VARIABLE U16 WITH VARIABLE I16
1309 MOVF BB1,W
1310 MOVWF TEMPY16
1311 MOVF BB1_H,W
1312 MOVWF TEMPY16_H
1313 CALL SUBV16
1314 ;SET VARIABLE I16 WITH VARIABLE U16
1315 BANKO
1316 MOVF TEMPX16,W
1317 MOVWF WARM
1318 MOVF TEMPX16_H,W
1319 MOVWF WARM_H
1320
1321 RETURN
1322
1323 LABEL_SET_SEGMENTS
1324
1325 ;OPERATION
1326 ;SET VARIABLE U16 WITH VARIABLE I16
1327 BANKO
1328 MOVF WARM,W
1329 MOVWF TEMPY16
1330 MOVF WARM_H,W
1331 MOVWF TEMPY16_H
1332 CALL ABSV16
1333 ;SET VARIABLE U16 WITH VARIABLE U16
1334 BANKO
1335 MOVF TEMPY16,W
1336 MOVWF NUMBER
1337 MOVF TEMPY16_H,W
1338 MOVWF NUMBER_H
1339

```

```

1340 ;SET VARIABLE U16 WITH CONSTANT
1341 MOVLW 0X64
1342 MOVWF DEC_PLACE
1343 MOVLW 0X0
1344 MOVWF DEC_PLACE_H
1345
1346 ;SET VARIABLE U16 WITH CONSTANT
1347 MOVLW 0X13
1348 MOVWF S_DATA_POINTER
1349 MOVLW 0X1
1350 MOVWF S_DATA_POINTER_H
1351
1352 ;BEGIN OF IF-STRUCTURE
1353 ;COMPARE-OPERATION
1354 ;SET VARIABLE U16 WITH VARIABLE I16
1355 MOVF WARM,W
1356 MOVWF TMPX16
1357 MOVF WARM_H,W
1358 MOVWF TMPX16_H
1359 ;SET VARIABLE U16 WITH CONSTANT
1360 MOVLW 0X0
1361 MOVWF TEMPY16
1362 MOVLW 0X0
1363 MOVWF TEMPY16_H
1364 CALL GRI16I16
1365 BANK0
1366 MOVWF RESULTS
1367 BTFSF RESULTS,0
1368 GOTO LABEL_1038
1369
1370 ;SET INDIRECT VARIABLE WITH CONSTANT
1371 ;SET U16 POINTER
1372 BCF STATUS,IRP
1373 BANK0
1374 BTFSF S_DATA_POINTER_H,0
1375 BSF STATUS,IRP
1376 BANK0
1377 MOVF S_DATA_POINTER,W
1378 MOVWF FSR
1379 MOVLW 0X40
1380 MOVWF INDF
1381
1382 GOTO LABEL_1039
1383 LABEL_1038
1384
1385 ;SET INDIRECT VARIABLE WITH CONSTANT
1386 ;SET U16 POINTER
1387 BCF STATUS,IRP
1388 BANK0
1389 BTFSF S_DATA_POINTER_H,0
1390 BSF STATUS,IRP
1391 BANK0
1392 MOVF S_DATA_POINTER,W
1393 MOVWF FSR
1394 MOVLW 0X0
1395 MOVWF INDF
1396
1397 LABEL_1039
1398 ;END OF IF-STRUCTURE
1399
1400 ;BEGIN OF FOR-LOOP
1401 ;SET VARIABLE U8 WITH CONSTANT
1402 MOVLW 0X3
1403 BANK0
1404 MOVWF J_DEC
1405 INCF J_DEC,F
1406 LABEL_1040
1407 BANK0
1408 DECF J_DEC,F
1409 BTFSF STATUS,Z
1410 GOTO LABEL_1041
1411
1412 ;SET VARIABLE U16 WITH VARIABLE U16
1413 BANK0
1414 MOVF NUMBER,W
1415 MOVWF TMPX16
1416 MOVF NUMBER_H,W
1417 MOVWF TMPX16_H
1418 ;OPERATION
1419 ;SET VARIABLE U16 WITH VARIABLE U16
1420 MOVF TMPX16,W
1421 MOVWF TEMPX16
1422 MOVF TMPX16_H,W
1423 MOVWF TEMPX16_H
1424 ;SET VARIABLE U16 WITH VARIABLE U16
1425 MOVF DEC_PLACE,W
1426 MOVWF TEMPY16
1427 MOVF DEC_PLACE_H,W
1428 MOVWF TEMPY16_H
1429 CALL DIVV16
1430 ;SET VARIABLE U16 WITH VARIABLE U16
1431 BANK0
1432 MOVF RESULT16,W
1433 MOVWF TMPX16
1434 MOVF RESULT16_H,W
1435 MOVWF TMPX16_H
1436
1437 ;SET VARIABLE U16 WITH VARIABLE U16
1438 MOVF TEMPX16,W
1439 MOVWF NUMBER
1440 MOVF TMPX16_H,W
1441 MOVWF NUMBER_H
1442 ;SET VARIABLE U8 WITH VARIABLE U16
1443 MOVF TMPX16,W
1444 MOVWF DEC
1445
1446 ;CALL LOOKUP-TABLE
1447 ;SET VARIABLE U8 WITH VARIABLE U8
1448 BANK0
1449 MOVF DEC,W
1450 MOVWF LOOKUP_TEMP
1451 CALL LABEL_LOOKUP_SEGMENT_CODES
1452 BANK0
1453 MOVWF DAT
1454
1455 ;BEGIN OF IF-STRUCTURE
1456 ;COMPARE-OPERATION
1457 ;SET VARIABLE U16 WITH VARIABLE U16
1458 MOVF DEC_PLACE,W

```

```

1459 MOVWF TEMPX16
1460 MOVF DEC_PLACE_H,W
1461 MOVWF TEMPX16_H
1462 ;SET VARIABLE U16 WITH CONSTANT
1463 MOVLW OXA
1464 MOVWF TEMPY16
1465 MOVLW OXO
1466 MOVWF TEMPY16_H
1467 CALL EQU16U16
1468 BANKO
1469 MOVWF RESULTS8
1470 BTFSCL RESULTS8,0
1471 GOTO LABEL_1042
1472
1473 GOTO LABEL_1043
1474 LABEL_1042
1475
1476 ;OPERATION
1477 ;SET VARIABLE U8 WITH VARIABLE U8
1478 BANKO
1479 MOVF DAT,W
1480 MOVWF TEMPX8
1481 ;SET VARIABLE U8 WITH CONSTANT
1482 MOVLW OX80
1483 MOVWF TEMPY8
1484 CALL ORV8
1485 ;SET VARIABLE U8 WITH VARIABLE U8
1486 BANKO
1487 MOVF TEMPX8,W
1488 MOVWF DAT
1489
1490 LABEL_1043
1491 ;END OF IF-STRUCTURE
1492
1493 ;BEGIN OF IF-STRUCTURE
1494 ;COMPARE-OPERATION
1495 ;SET VARIABLE U16 WITH VARIABLE U16
1496 BANKO
1497 MOVF DEC_PLACE,W
1498 MOVWF TEMPX16
1499 MOVF DEC_PLACE_H,W
1500 MOVWF TEMPX16_H
1501 ;SET VARIABLE U16 WITH CONSTANT
1502 MOVLW OX64
1503 MOVWF TEMPY16
1504 MOVLW OXO
1505 MOVWF TEMPY16_H
1506 CALL EQU16U16
1507 BANKO
1508 MOVWF RESULTS8
1509 BTFSCL RESULTS8,0
1510 GOTO LABEL_1044
1511
1512 GOTO LABEL_1047
1513 LABEL_1044
1514
1515 ;BEGIN OF IF-STRUCTURE
1516 ;COMPARE-OPERATION
1517 ;SET VARIABLE U8 WITH VARIABLE U8
1518 BANKO
1519 MOVF DEC,W
1520 MOVWF TEMPX8
1521 ;SET VARIABLE U8 WITH CONSTANT
1522 MOVLW OXO
1523 MOVWF TEMPY8
1524 CALL EQU8U8
1525 BANKO
1526 MOVWF RESULTS8
1527 BTFSCL RESULTS8,0
1528 GOTO LABEL_1045
1529
1530 GOTO LABEL_1046
1531 LABEL_1045
1532
1533 BANKO
1534 CLRWF DAT
1535
1536 LABEL_1046
1537 ;END OF IF-STRUCTURE
1538
1539 LABEL_1047
1540 ;END OF IF-STRUCTURE
1541
1542 BANKO
1543 MOVF S_DATA_POINTER,F
1544 BTFSCL STATUS,Z
1545 DECF S_DATA_POINTER_H,F
1546 BANKO
1547 DECF S_DATA_POINTER,F
1548
1549 ;SET INDIRECT VARIABLE WITH VARIABLE U8
1550 ;SET U16 POINTER
1551 BCF STATUS,IRP
1552 BTFSCL S_DATA_POINTER_H,0
1553 BSF STATUS,IRP
1554 BANKO
1555 MOVF S_DATA_POINTER,W
1556 MOVWF FSR
1557 MOVF DAT,W
1558 MOVWF INDF
1559
1560 ;OPERATION
1561 ;SET VARIABLE U16 WITH VARIABLE U16
1562 MOVF DEC_PLACE,W
1563 MOVWF TEMPX16
1564 MOVF DEC_PLACE_H,W
1565 MOVWF TEMPX16_H
1566 ;SET VARIABLE U16 WITH CONSTANT
1567 MOVLW OXA
1568 MOVWF TEMPY16
1569 MOVLW OXO
1570 MOVWF TEMPY16_H
1571 CALL DIVV16
1572 ;SET VARIABLE U16 WITH VARIABLE U16
1573 BANKO
1574 MOVF RESULT16,W
1575 MOVWF DEC_PLACE
1576 MOVF RESULT16_H,W
1577 MOVWF DEC_PLACE_H

```

```

1578
1579     GOTO LABEL_1040
1580 LABEL_1041
1581 ;END OF FOR/WHILE-LOOP
1582
1583     RETURN
1584
1585 OUT_GENERIC_PORT1
1586     BANK0
1587     BCF     PORTA ,3
1588     BTFS   TEMPPTS ,0
1589     BSF     PORTA ,3
1590     BCF     PORTA ,2
1591     BTFS   TEMPPTS ,1
1592     BSF     PORTA ,2
1593     BCF     PORTA ,1
1594     BTFS   TEMPPTS ,2
1595     BSF     PORTA ,1
1596     BCF     PORTA ,0
1597     BTFS   TEMPPTS ,3
1598     BSF     PORTA ,0
1599     RETURN
1600
1601 ;*****LOOKUP-TABLES*****
1602
1603     ORG 0X4EC
1604 LABEL_LOOKUP_DIGIT_SELECT
1605
1606 ;BEGIN OF IF-STRUCTURE
1607 ;COMPARE-OPERATION
1608 ;SET VARIABLE US WITH VARIABLE US
1609     BANK0
1610     MOVF   LOOKUP_TEMP,W
1611     MOVWF TEMPX8
1612 ;SET VARIABLE US WITH CONSTANT
1613     MOVLW 0X3
1614     MOVWF TEMPY8
1615     CALL  GRU8U8
1616     BANK0
1617     MOVWF RESULTS
1618     BTFS   RESULTS ,0
1619     GOTO  LABEL_1005
1620
1621     RETLW 0X0
1622
1623     GOTO LABEL_1006
1624 LABEL_1005
1625
1626 LABEL_1006
1627 ;END OF IF-STRUCTURE
1628
1629     MOVLW 0X5
1630     MOVWF PCLATH
1631     BANK0
1632     MOVF   LOOKUP_TEMP,W
1633     ADDWF PCL
1634     RETLW 0XE
1635     RETLW 0XD
1636     RETLW 0XB
1637     RETLW 0X7
1638
1639     ORG 0X53C
1640 LABEL_LOOKUP_SEGMENT_CODES
1641
1642 ;BEGIN OF IF-STRUCTURE
1643 ;COMPARE-OPERATION
1644 ;SET VARIABLE US WITH VARIABLE US
1645     BANK0
1646     MOVF   LOOKUP_TEMP,W
1647     MOVWF TEMPX8
1648 ;SET VARIABLE US WITH CONSTANT
1649     MOVLW 0XA
1650     MOVWF TEMPY8
1651     CALL  GRU8U8
1652     BANK0
1653     MOVWF RESULTS
1654     BTFS   RESULTS ,0
1655     GOTO  LABEL_1007
1656
1657     RETLW 0X0
1658
1659     GOTO LABEL_1008
1660 LABEL_1007
1661
1662 LABEL_1008
1663 ;END OF IF-STRUCTURE
1664
1665     MOVLW 0X5
1666     MOVWF PCLATH
1667     BANK0
1668     MOVF   LOOKUP_TEMP,W
1669     ADDWF PCL
1670     RETLW 0X3F
1671     RETLW 0X6
1672     RETLW 0X5B
1673     RETLW 0X4F
1674     RETLW 0X66
1675     RETLW 0X6D
1676     RETLW 0X7D
1677     RETLW 0X7
1678     RETLW 0X7F
1679     RETLW 0X6F
1680     RETLW 0X40
1681
1682 ;*****INTERRUPT SERVICE ROUTINE*****
1683
1684 LABEL_ISR
1685     MOVWF 0X7F
1686     SWAPF STATUS,W
1687     BANK0
1688     MOVWF ISR_STATUS
1689     MOVF  0X7F,W
1690     MOVWF ISR_W
1691     MOVF  TEMPPTS,W
1692     MOVWF ISR_TEMPPTS
1693     MOVF  FSR,W
1694     MOVWF ISR_FSR
1695     BANK1
1696     BTFSS PIE1 ,TMR2IE

```



```

1697 GOTO LABEL_1072
1698 BANKO
1699 BTFSS PIR1,TMR2IF
1700 GOTO LABEL_1072
1701 CALL LABEL_EVENT0
1702 BANKO
1703 BCF PIR1,TMR2IF
1704 LABEL_1072
1705 BANK1
1706 BTFSS PIE1,TMR1IE
1707 GOTO LABEL_1073
1708 BANKO
1709 BTFSS PIR1,TMR1IF
1710 GOTO LABEL_1073
1711 CALL LABEL_EVENT1
1712 BANKO
1713 BCF PIR1,TMR1IF
1714 LABEL_1073
1715 BANKO
1716 MOVF ISR_FSR,W
1717 MOVWF FSR
1718 MOVF ISR_TEMPPORT8,W
1719 MOVWF TEMPPORT8
1720 MOVF ISR_W,W
1721 MOVWF 0X7F
1722 SWAPF ISR_STATUS,W
1723 MOVWF STATUS
1724 SWAPF 0X7F,F
1725 SWAPF 0X7F,W
1726 RETFIE
1727
1728 ;*****EVENT-ROUTINES*****
1729
1730 LABEL_EVENT0
1731
1732 ;BEGIN OF IF-STRUCTURE
1733 ;COMPARE-OPERATION
1734 ;SET VARIABLE U8 WITH VARIABLE U8
1735 BANKO
1736 MOVF DISPLAY_STATE,W
1737 MOVWF ISR_TEMPX8
1738 ;SET VARIABLE U8 WITH CONSTANT
1739 MOVLW 0X0
1740 MOVWF ISR_TEMPY8
1741 CALL ISR_EQU8U8
1742 BANKO
1743 MOVWF ISR_RESULTS
1744 BTFSC ISR_RESULTS,0
1745 GOTO LABEL_1012
1746
1747 ;BEGIN OF IF-STRUCTURE
1748 ;COMPARE-OPERATION
1749 ;SET VARIABLE U8 WITH VARIABLE U8
1750 BANKO
1751 MOVF DISPLAY_STATE,W
1752 MOVWF ISR_TEMPX8
1753 ;SET VARIABLE U8 WITH CONSTANT
1754 MOVLW 0X1
1755 MOVWF ISR_TEMPY8
1756 CALL ISR_EQU8U8
1757 BANKO
1758 MOVWF ISR_RESULTS
1759 BTFSC ISR_RESULTS,0
1760 GOTO LABEL_1013
1761
1762 ;BEGIN OF IF-STRUCTURE
1763 ;COMPARE-OPERATION
1764 ;SET VARIABLE U8 WITH VARIABLE U8
1765 BANKO
1766 MOVF DISPLAY_STATE,W
1767 MOVWF ISR_TEMPX8
1768 ;SET VARIABLE U8 WITH CONSTANT
1769 MOVLW 0X2
1770 MOVWF ISR_TEMPY8
1771 CALL ISR_EQU8U8
1772 BANKO
1773 MOVWF ISR_RESULTS
1774 BTFSC ISR_RESULTS,0
1775 GOTO LABEL_1014
1776
1777 ;BEGIN OF IF-STRUCTURE
1778 ;COMPARE-OPERATION
1779 ;SET VARIABLE U8 WITH VARIABLE U8
1780 BANKO
1781 MOVF DISPLAY_STATE,W
1782 MOVWF ISR_TEMPX8
1783 ;SET VARIABLE U8 WITH CONSTANT
1784 MOVLW 0X3
1785 MOVWF ISR_TEMPY8
1786 CALL ISR_EQU8U8
1787 BANKO
1788 MOVWF ISR_RESULTS
1789 BTFSC ISR_RESULTS,0
1790 GOTO LABEL_1015
1791
1792 ;BEGIN OF IF-STRUCTURE
1793 ;COMPARE-OPERATION
1794 ;SET VARIABLE U8 WITH VARIABLE U8
1795 BANKO
1796 MOVF DISPLAY_STATE,W
1797 MOVWF ISR_TEMPX8
1798 ;SET VARIABLE U8 WITH CONSTANT
1799 MOVLW 0X4
1800 MOVWF ISR_TEMPY8
1801 CALL ISR_EQU8U8
1802 BANKO
1803 MOVWF ISR_RESULTS
1804 BTFSC ISR_RESULTS,0
1805 GOTO LABEL_1018
1806
1807 GOTO LABEL_1023
1808 LABEL_1018
1809
1810 ;SET SINGLE OUPUT PIN
1811 BANKO
1812 BSF PORTB,0
1813
1814 ;SET VIRTUAL PORT VALUE WITH VARIABLE U8
1815 MOVF DIGIT_BLUE_CODE,W

```

```

1816 MOVWF TEMPPORT8
1817 CALL OUT_GENERIC_PORT1
1818
1819 BANK0
1820 INCF T_BCD_POINTER,F
1821 BTFS STATUS,Z
1822 INCF T_BCD_POINTER_H,F
1823
1824 BANK0
1825 INCF DIGIT_INDEX,F
1826
1827 ;BEGIN OF IF-STRUCTURE
1828 ;COMPARE-OPERATION
1829 ;SET VARIABLE US WITH VARIABLE US
1830 MOVF DIGIT_INDEX,W
1831 MOVWF ISR_TEMP8
1832 ;SET VARIABLE US WITH CONSTANT
1833 MOVLW 0X3
1834 MOVWF ISR_TEMP8
1835 CALL ISR_GRU8U8
1836 BANK0
1837 MOVWF ISR_RESULTS
1838 BTFS ISR_RESULTS,0
1839 GOTO LABEL_1020
1840
1841 BANK0
1842 CLRF DIGIT_INDEX
1843
1844 ;SET VARIABLE U16 WITH CONSTANT
1845 MOVLW 0X10
1846 MOVWF T_BCD_POINTER
1847 MOVLW 0X1
1848 MOVWF T_BCD_POINTER_H
1849
1850 GOTO LABEL_1022
1851 LABEL_1020
1852
1853 LABEL_1022
1854 ;END OF IF-STRUCTURE
1855
1856 ;SET VARIABLE US WITH INDIRECT VARIABLE
1857 ;SET U16 POINTER
1858 BCF STATUS,IRP
1859 BANK0
1860 BTFS T_BCD_POINTER_H,0
1861 BSF STATUS,IRP
1862 BANK0
1863 MOVF T_BCD_POINTER,W
1864 MOVWF FSR
1865 MOVF INDF,W
1866 MOVWF CURRENT_DIGIT_CODE
1867
1868 ;CALL LOOKUP-TABLE
1869 ;SET VARIABLE US WITH VARIABLE US
1870 BANK0
1871 MOVF DIGIT_INDEX,W
1872 MOVWF LOOKUP_TEMP
1873 CALL LABEL_LOOKUP_DIGIT_SELECT
1874 BANK0
1875 MOVWF DIGIT_BLUE_CODE
1876
1877 ;SET VARIABLE US WITH CONSTANT
1878 MOVLW 0X0
1879 MOVWF DISPLAY_STATE
1880
1881 LABEL_1023
1882 ;END OF IF-STRUCTURE
1883
1884 GOTO LABEL_1024
1885 LABEL_1015
1886
1887 ;SET SINGLE OUPUT PIN
1888 BANK0
1889 BSF PORTB,1
1890
1891 INCF SEGMENT_INDEX,F
1892
1893 ;BEGIN OF IF-STRUCTURE
1894 ;COMPARE-OPERATION
1895 ;SET VARIABLE US WITH VARIABLE US
1896 MOVF SEGMENT_INDEX,W
1897 MOVWF ISR_TEMP8
1898 ;SET VARIABLE US WITH CONSTANT
1899 MOVLW 0X7
1900 MOVWF ISR_TEMP8
1901 CALL ISR_GRU8U8
1902 BANK0
1903 MOVWF ISR_RESULTS
1904 BTFS ISR_RESULTS,0
1905 GOTO LABEL_1019
1906
1907 BANK0
1908 CLRF SEGMENT_INDEX
1909
1910 ;SET VARIABLE US WITH CONSTANT
1911 MOVLW 0X4
1912 MOVWF DISPLAY_STATE
1913
1914 GOTO LABEL_1021
1915 LABEL_1019
1916
1917 ;SET VARIABLE US WITH CONSTANT
1918 MOVLW 0X1
1919 BANK0
1920 MOVWF DISPLAY_STATE
1921
1922 LABEL_1021
1923 ;END OF IF-STRUCTURE
1924
1925 LABEL_1024
1926 ;END OF IF-STRUCTURE
1927
1928 GOTO LABEL_1025
1929 LABEL_1014
1930
1931 ;BEGIN OF IF-STRUCTURE (DEPENDING ON BIT/PIN)
1932 BANK0
1933 BTFS CURRENT_DIGIT_CODE,7
1934 GOTO LABEL_1016

```

```

1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
;SET SINGLE OUPUT PIN
BANKO
BSF PORTA,4

GOTO LABEL_1017
LABEL_1016

;SET SINGLE OUPUT PIN
BANKO
BCF PORTA,4

LABEL_1017
;END OF IF-STRUCTURE

;OPERATION
;SET VARIABLE US WITH VARIABLE US
BANKO
MOVF CURRENT_DIGIT_CODE,W
MOVWF ISR_TEMPY8
CALL ISR_SHLV8
;SET VARIABLE US WITH VARIABLE US
BANKO
MOVF ISR_TEMPY8,W
MOVWF CURRENT_DIGIT_CODE

;SET VARIABLE US WITH CONSTANT
MOVLW 0X3
MOVWF DISPLAY_STATE

LABEL_1025
;END OF IF-STRUCTURE

GOTO LABEL_1026
LABEL_1013

;SET SINGLE OUPUT PIN
BANKO
BCF PORTB,1

;SET VARIABLE US WITH CONSTANT
MOVLW 0X2
MOVWF DISPLAY_STATE

LABEL_1026
;END OF IF-STRUCTURE

GOTO LABEL_1027
LABEL_1012

;SET SINGLE OUPUT PIN
BANKO
BCF PORTB,0

;SET VARIABLE US WITH CONSTANT
MOVLW 0X1
MOVWF DISPLAY_STATE

LABEL_1027
;END OF IF-STRUCTURE

RETURN

LABEL_EVENT1

BANKO
INCF MY_TIMER,F
BTFSZ STATUS,Z
INCF MY_TIMER_H,F

;BEGIN OF IF-STRUCTURE
;COMPARE-OPERATION
;SET VARIABLE U16 WITH VARIABLE U16
BANKO
MOVF MY_TIMER,W
MOVWF ISR_TEMPX16
MOVWF MY_TIMER_H,W
MOVWF ISR_TEMPX16_H
;SET VARIABLE U16 WITH CONSTANT
MOVLW 0X86
MOVWF ISR_TEMPY16
MOVLW 0XD
MOVWF ISR_TEMPY16_H
CALL ISR_GRU16U16
BANKO
MOVWF ISR_RESULTS8
BTFSZ ISR_RESULTS8,0
GOTO LABEL_1048

BANKO
CLRF MY_TIMER
CLRF MY_TIMER_H

;SET SINGLE OUPUT PIN
BCF PORTB,3

GOTO LABEL_1049
LABEL_1048

LABEL_1049
;END OF IF-STRUCTURE

RETURN

END

```