Akademia
Techniczno-Humanistyczna
w Bielsku-Białej

Department of machine construction and computer science

Analog and digital electronic circuits

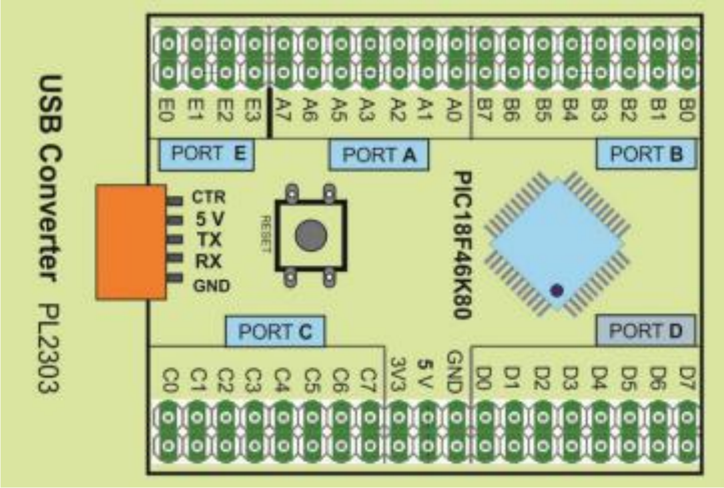
7 Segment Display Interfacing with PIC Microcontroller (PIC18)

Adrian RUTKOWSKI
Marcin WÓJCIK

1. Description

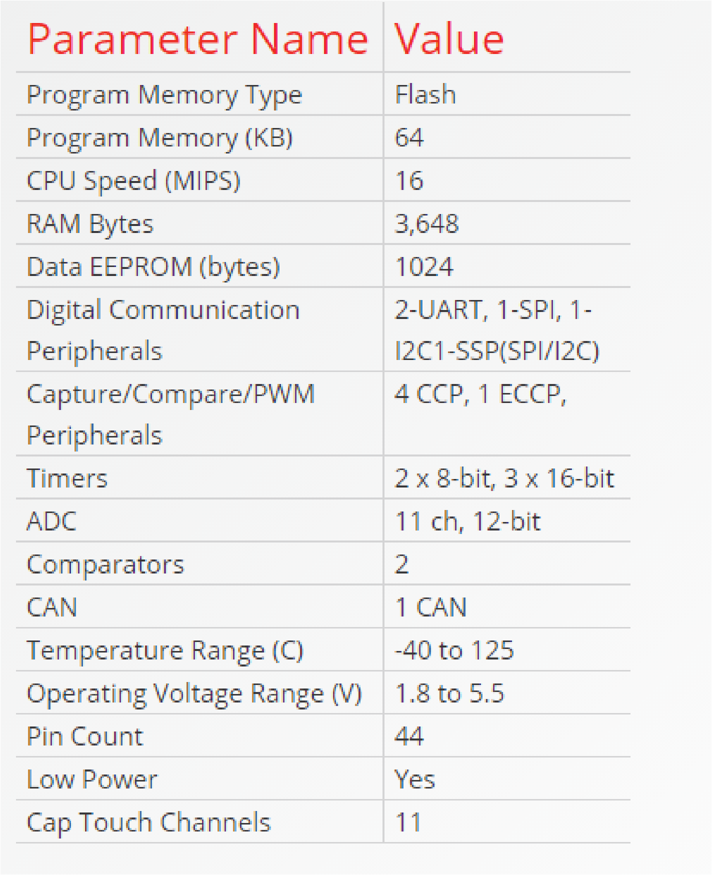
The project presents a 7 segment display using a PIC microcontroller (PIC18). The program was written in MPLAB environment. The program was thrown into the microprocessor using the bootloader program. The program allows to count the pulses with a button and display them graphically with a 7 segment display.

The project was tested on a development board BOOTLOADER BEL8


USB Converter PL2303
E0 E1 E2 E3 A7 A6 A5 A4 A3 A2 A1 A0 B7 B6 B5 B4 B3 B2 B1 B0
PORT E
PORT A
PORT B
CTR
5 V
TX
RX
GND
RESET
PIC18F46K80
PORT C
PORT D
C0 C1 C2 C3 C4 C5 C6 C7 3V3 5 V GND D0 D1 D2 D3 D4 D5 D6 D7

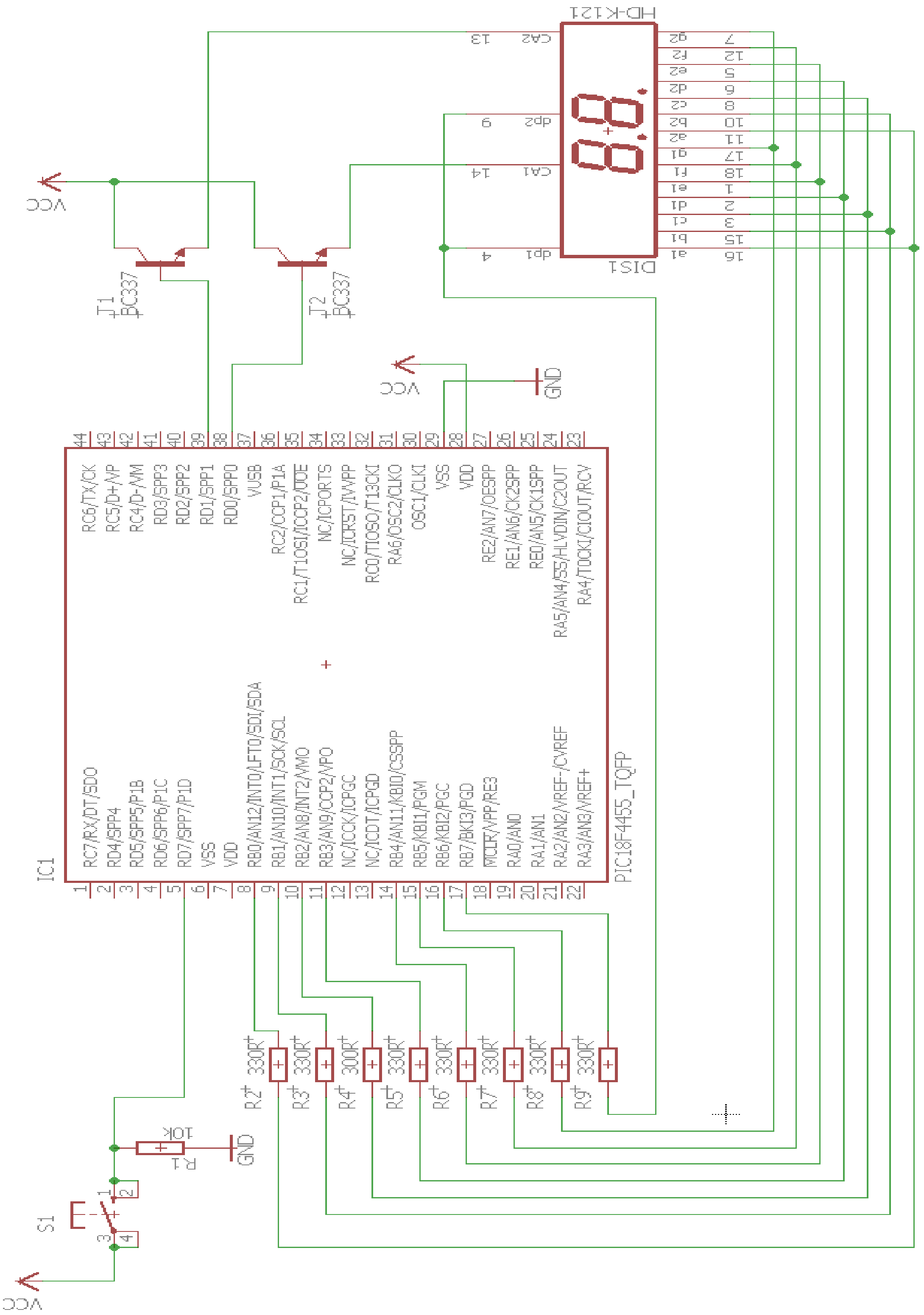
Features PIC18F46K80:

- Featuring nanoWatt XLP Technology ideal for battery applications
- 20 nA Sleep mode (Watch Webinar)
- 300nA Watch Dog Timer operation in Sleep modes
- Integrated ECAN™ conforming to CAN 2.0B active specification
- Extreme Low Power Sleep mode ideal for battery applications
- Operating voltage 1.8V – 5.5V
- Charge Time Measurement Unit (CTMU) supports Touch sensing and advanced sensing applications
- 11 ch, 12-bit ADC
- 4 Capture / Compare / PWM modules and 1 Enhanced Capture / Compare / PWM modules
- 1 MSSP serial ports for SPI or I2C™ communication
- 2 Enhanced USART modules
- Two analog comparators
- Digital Signal Modulator
- Self programming Flash supports 10k erase/write cycles & 20 years retention



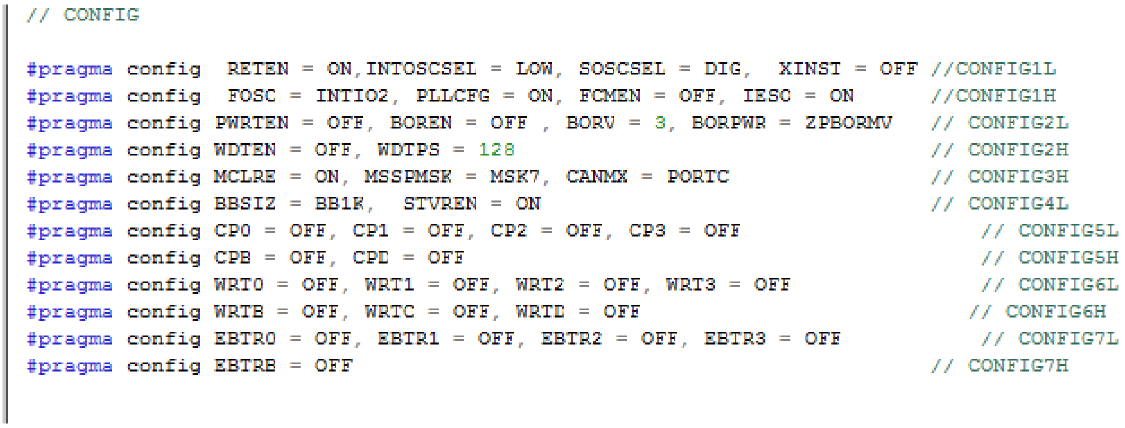
Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	64
CPU Speed (MIPS)	16
RAM Bytes	3,648
Data EEPROM (bytes)	1024
Digital Communication Peripherals	2-UART, 1-SPI, 1-I2C1-SSP(SPI/I2C)
Capture/Compare/PWM Peripherals	4 CCP, 1 ECCP,
Timers	2 x 8-bit, 3 x 16-bit
ADC	11 ch, 12-bit
Comparators	2
CAN	1 CAN
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	44
Low Power	Yes
Cap Touch Channels	11

2. Scheme


HD-K121
DIS1
CA2
13
CA1
14
dp2
9
dp1
4
g2
7
f2
12
e2
5
d2
6
c2
8
b2
10
a2
11
g1
17
f1
18
e1
1
d1
2
c1
3
b1
15
a1
16
VCC
T1
BC337
T2
BC337
VCC
GND
IC1
PIC18F4455_TQFP
RC7/RX/DT/SDO
RD4/SPP4
RD5/SPP5/P1B
RD6/SPP6/P1C
RD7/SPP7/P1D
VSS
VDD
RB0/AN12/INT0/LFT0/SDI/SDA
RB1/AN10/INT1/SCK/SCL
RB2/AN8/INT2/VMO
RB3/AN9/CCP2/VPO
NC/ICCK/ICPGC
NC/ICDT/ICPGD
RB4/AN11/KBI0/CSSPP
RB5/KBI1/PGM
RB6/KBI2/PGC
RB7/KBI3/PGD
MCLR/VPP/RE3
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RC6/TX/CK
RC5/D+/VP
RC4/D-/VM
RD3/SPP3
RD2/SPP2
RD1/SPP1
RD0/SPP0
VUSB
RC2/CCP1/P1A
RC1/T1OSI/CCP2/UOE
NC/ICPORTS
NC/ICRST/ICVPP
RC0/T1OSO/T13CKI
RA6/OSC2/CLKO
OSC1/CLKI
VSS
VDD
RE2/AN7/OESPP
RE1/AN6/CK2SPP
RE0/AN5/CK1SPP
RA5/AN4/SS/HLVDIN/C2OUT
RA4/T0CKI/C1OUT/RCV
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
R2
330R
R3
330R
R4
300R
R5
330R
R6
330R
R7
330R
R8
330R
R9
330R
R1
10K
GND
S1
1
2
3
4
VCC

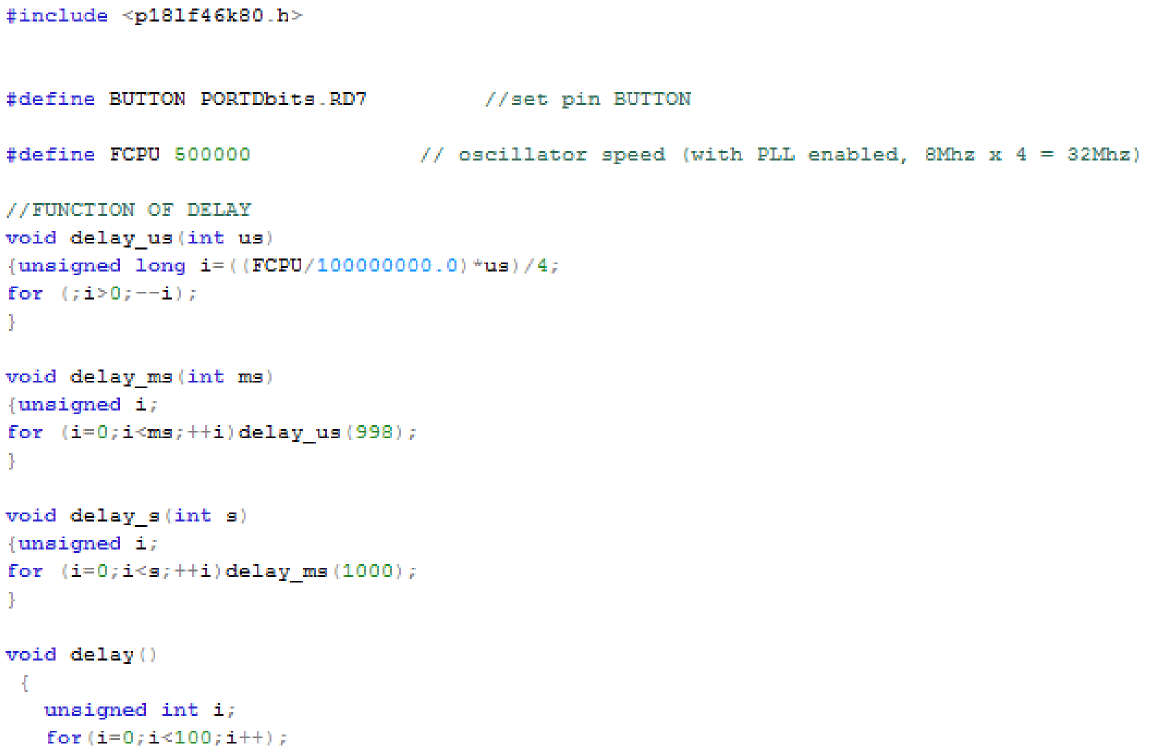
3. Program

3.1 Setting the configuration parameters of the microcontroller used:



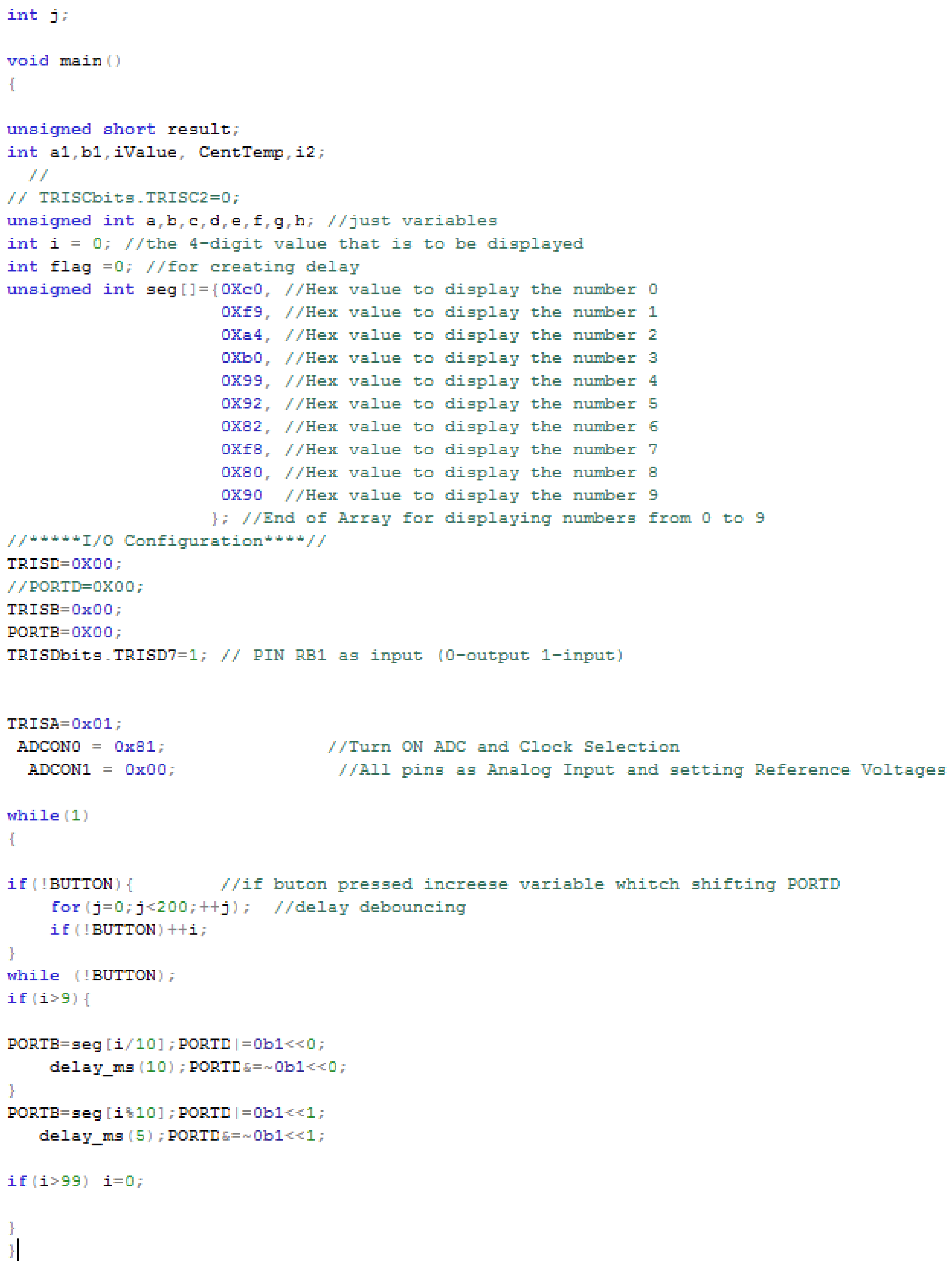
```
// CONFIG

#pragma config  RETEN = ON,INTOSCSEL = LOW, SOSCSEL = DIG,  XINST = OFF //CONFIG1L
#pragma config  FOSC = INTIO2, PLLCFG = ON, FCMEN = OFF, IESO = ON      //CONFIG1H
#pragma config PWRTEN = OFF, BOREN = OFF , BORV = 3, BORPWR = ZPBORMV   // CONFIG2L
#pragma config WDTEN = OFF, WDTPS = 128                                 // CONFIG2H
#pragma config MCLRE = ON, MSSPMSK = MSK7, CANMX = PORTC                // CONFIG3H
#pragma config BBSIZ = BB1K,  STVREN = ON                               // CONFIG4L
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF                   // CONFIG5L
#pragma config CPB = OFF, CPD = OFF                                         // CONFIG5H
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF               // CONFIG6L
#pragma config WRTB = OFF, WRTC = OFF, WRTD = OFF                          // CONFIG6H
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF           // CONFIG7L
#pragma config EBTRB = OFF                                              // CONFIG7H
```

3.2 Definition of the pin to which the button has been connected according to the scheme and delay functions.



```
#include <p18lf46k80.h>


#define BUTTON PORTDbits.RD7         //set pin BUTTON

#define FCPU 500000             // oscillator speed (with PLL enabled, 8Mhz x 4 = 32Mhz)

//FUNCTION OF DELAY
void delay_us(int us)
{unsigned long i=((FCPU/100000000.0)*us)/4;
for (;i>0;--i);
}

void delay_ms(int ms)
{unsigned i;
for (i=0;i<ms;++i)delay_us(998);
}

void delay_s(int s)
{unsigned i;
for (i=0;i<s;++i)delay_ms(1000);
}

void delay()
 {
   unsigned int i;
   for(i=0;i<100;i++);
```

3.3 The main loop of the program



```
int j;

void main()
{

unsigned short result;
int a1,b1,iValue, CentTemp,i2;
  //
// TRISCbits.TRISC2=0;
unsigned int a,b,c,d,e,f,g,h; //just variables
int i = 0; //the 4-digit value that is to be displayed
int flag =0; //for creating delay
unsigned int seg[]={0Xc0, //Hex value to display the number 0
                    0Xf9, //Hex value to display the number 1
                    0Xa4, //Hex value to display the number 2
                    0Xb0, //Hex value to display the number 3
                    0X99, //Hex value to display the number 4
                    0X92, //Hex value to display the number 5
                    0X82, //Hex value to display the number 6
                    0Xf8, //Hex value to display the number 7
                    0X80, //Hex value to display the number 8
                    0X90  //Hex value to display the number 9
                   }; //End of Array for displaying numbers from 0 to 9
//*****I/O Configuration****//
TRISD=0X00;
//PORTD=0X00;
TRISB=0x00;
PORTB=0X00;
TRISDbits.TRISD7=1; // PIN RB1 as input (0-output 1-input)


TRISA=0x01;
 ADCON0 = 0x81;                //Turn ON ADC and Clock Selection
  ADCON1 = 0x00;                //All pins as Analog Input and setting Reference Voltages

while(1)
{

if(!BUTTON){        //if buton pressed increese variable whitch shifting PORTD
    for(j=0;j<200;++j);  //delay debouncing
    if(!BUTTON)++i;
}
while (!BUTTON);
if(i>9){

PORTB=seg[i/10];PORTD|=0b1<<0;
    delay_ms(10);PORTD&=~0b1<<0;
}
PORTB=seg[i%10];PORTD|=0b1<<1;
   delay_ms(5);PORTD&=~0b1<<1;

if(i>99) i=0;

}
}
```

The first part of the program set low states on the ports to which the 7-channel display was connected. There is also an array of characters from 0-9 that makes it easy to display specific characters on the display. In addition, as input, the pin responsible for the button is inserted. In the main program loop we wait for the button to be pressed. Each press is incremented and displayed on the display. When the maximum value is exceeded, the counter is reset.

We have used multiplexing to prevent the premature firing of segments.

4. Summary

The theme of the project was to create a schematic and program for the MICROCHIP PIC18 microcontroller using BOOTLOADER BEL8 development board. The task was to graphically display counts using the 7 segment display. The program was written entirely in MBPLAB environment using C language. With the implementation of 7 segment display, it is easy to expand the program with additional capabilities.