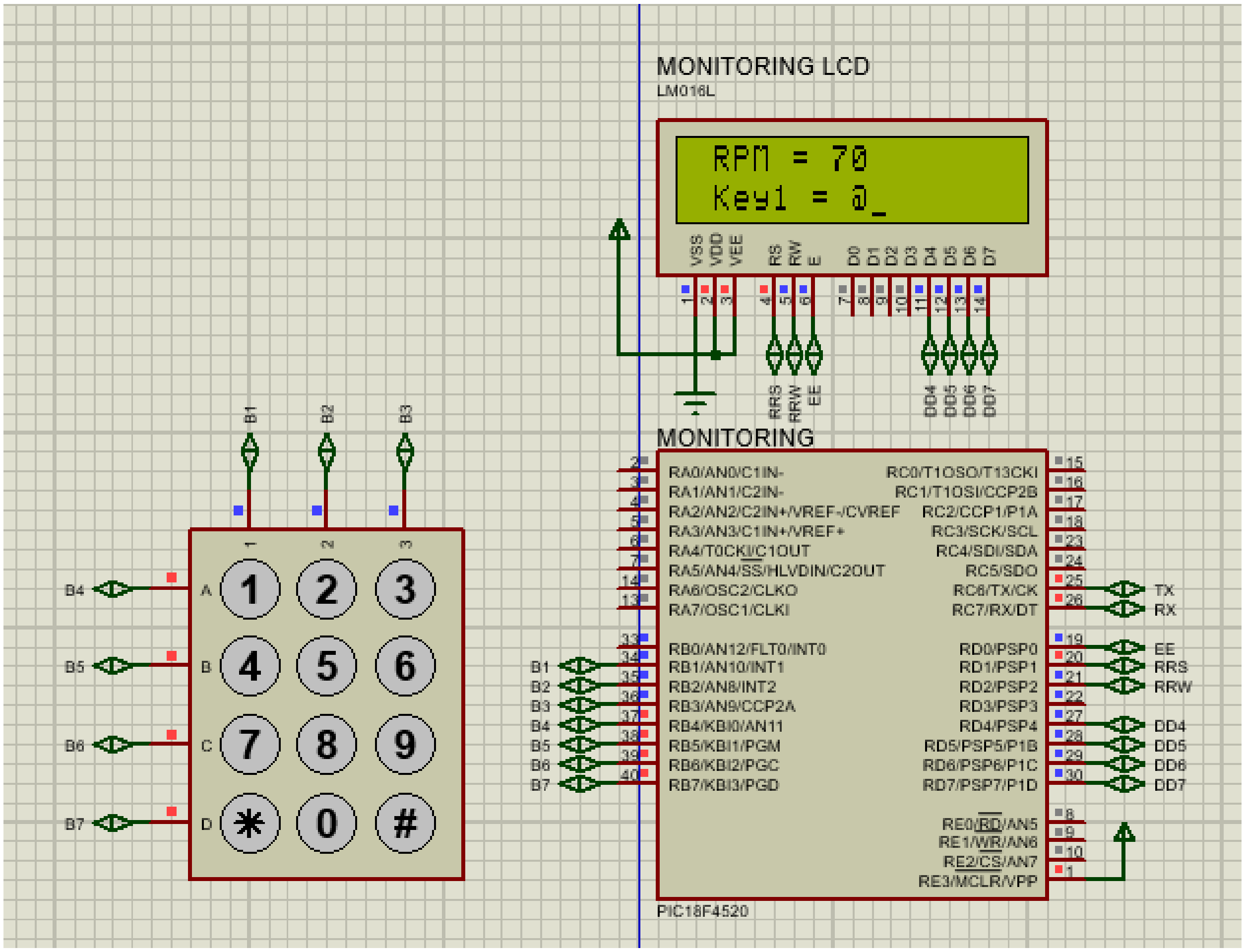
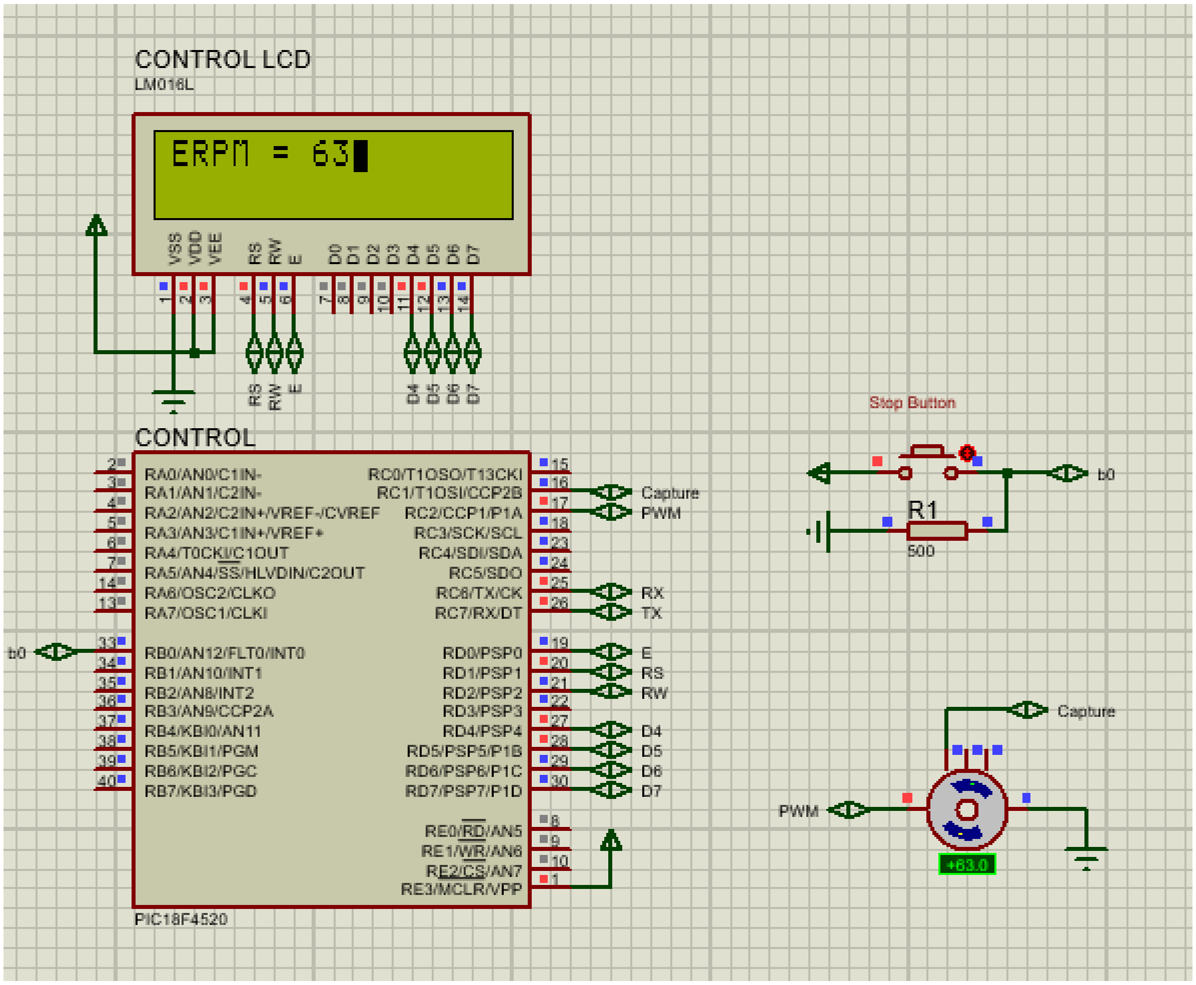
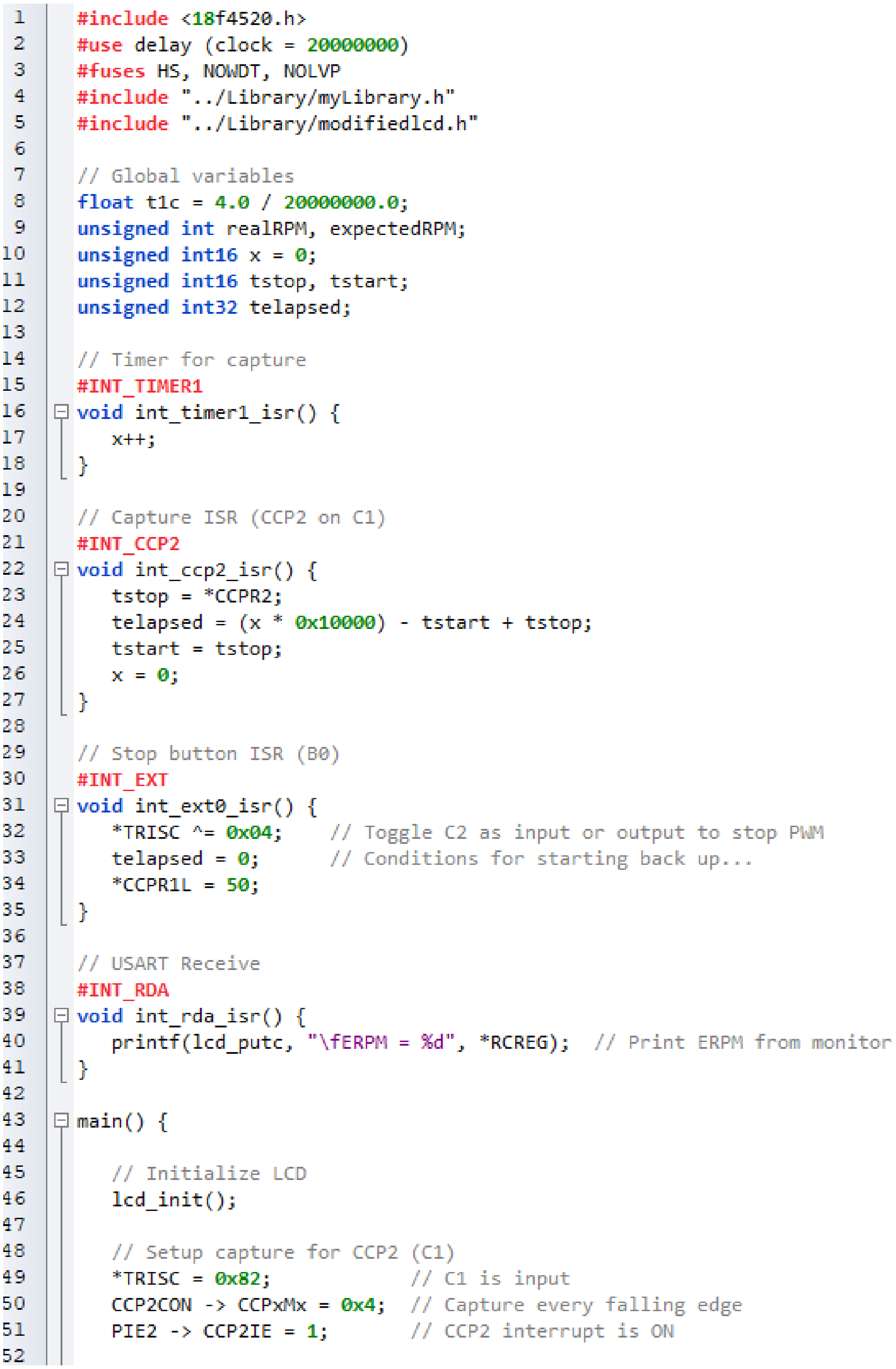
Final Project – USART Cruise Control – James Martin

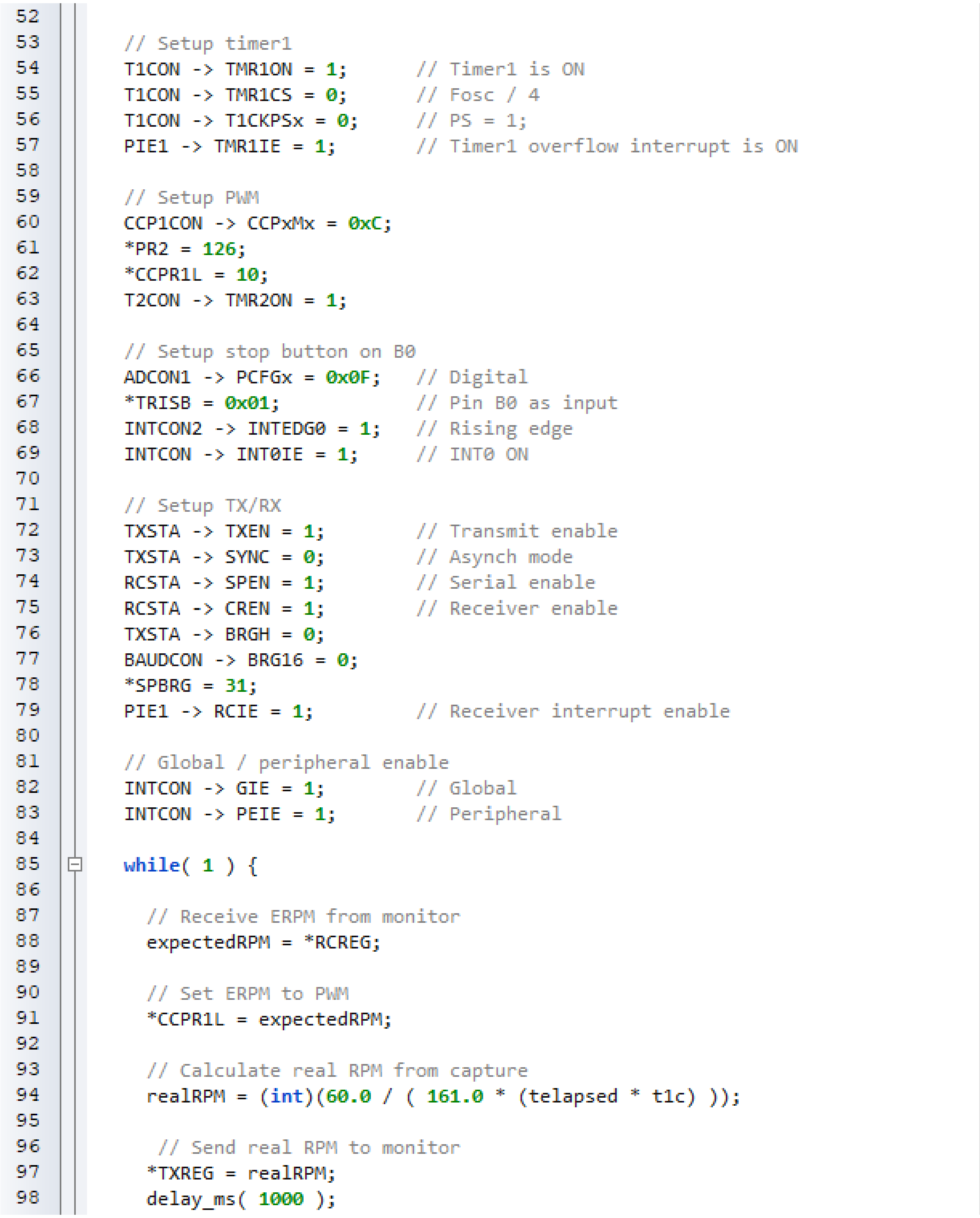



CONTROL LCD
LM016L
ERPM = 63
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
RS
RW
E
D4
D5
D6
D7
CONTROL
2
RA0/AN0/C1IN-
3
RA1/AN1/C2IN-
4
RA2/AN2/C2IN+/VREF-/CVREF
5
RA3/AN3/C1IN+/VREF+
6
RA4/T0CKI/C1OUT
7
RA5/AN4/SS/HLVDIN/C2OUT
14
RA6/OSC2/CLKO
13
RA7/OSC1/CLKI
b0
33
RB0/AN12/FLT0/INT0
34
RB1/AN10/INT1
35
RB2/AN8/INT2
36
RB3/AN9/CCP2A
37
RB4/KBI0/AN11
38
RB5/KBI1/PGM
39
RB6/KBI2/PGC
40
RB7/KBI3/PGD
RC0/T1OSO/T13CKI
15
RC1/T1OSI/CCP2B
16
Capture
RC2/CCP1/P1A
17
PWM
RC3/SCK/SCL
18
RC4/SDI/SDA
23
RC5/SDO
24
RC6/TX/CK
25
RX
RC7/RX/DT
26
TX
RD0/PSP0
19
E
RD1/PSP1
20
RS
RD2/PSP2
21
RW
RD3/PSP3
22
RD4/PSP4
27
D4
RD5/PSP5/P1B
28
D5
RD6/PSP6/P1C
29
D6
RD7/PSP7/P1D
30
D7
RE0/RD/AN5
8
RE1/WR/AN6
9
RE2/CS/AN7
10
RE3/MCLR/VPP
1
PIC18F4520
Stop Button
b0
R1
500
Capture
PWM
+63.0

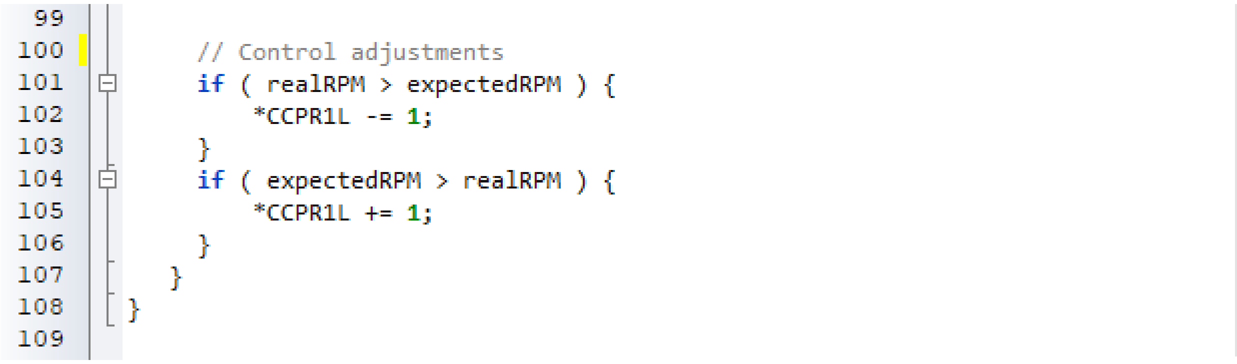
CONTROL:



```
#include <18f4520.h>
#use delay (clock = 20000000)
#fuses HS, NOWDT, NOLVP
#include "../Library/myLibrary.h"
#include "../Library/modifiedlcd.h"

// Global variables
float t1c = 4.0 / 20000000.0;
unsigned int realRPM, expectedRPM;
unsigned int16 x = 0;
unsigned int16 tstop, tstart;
unsigned int32 telapsed;

// Timer for capture
#INT_TIMER1
void int_timer1_isr() {
   x++;
}

// Capture ISR (CCP2 on C1)
#INT_CCP2
void int_ccp2_isr() {
   tstop = *CCPR2;
   telapsed = (x * 0x10000) - tstart + tstop;
   tstart = tstop;
   x = 0;
}

// Stop button ISR (B0)
#INT_EXT
void int_ext0_isr() {
   *TRISC ^= 0x04;    // Toggle C2 as input or output to stop PWM
   telapsed = 0;      // Conditions for starting back up...
   *CCPR1L = 50;
}

// USART Receive
#INT_RDA
void int_rda_isr() {
   printf(lcd_putc, "\fERPM = %d", *RCREG);   // Print ERPM from monitor
}

main() {

   // Initialize LCD
   lcd_init();

   // Setup capture for CCP2 (C1)
   *TRISC = 0x82;            // C1 is input
   CCP2CON -> CCPxMx = 0x4;  // Capture every falling edge
   PIE2 -> CCP2IE = 1;       // CCP2 interrupt is ON

```

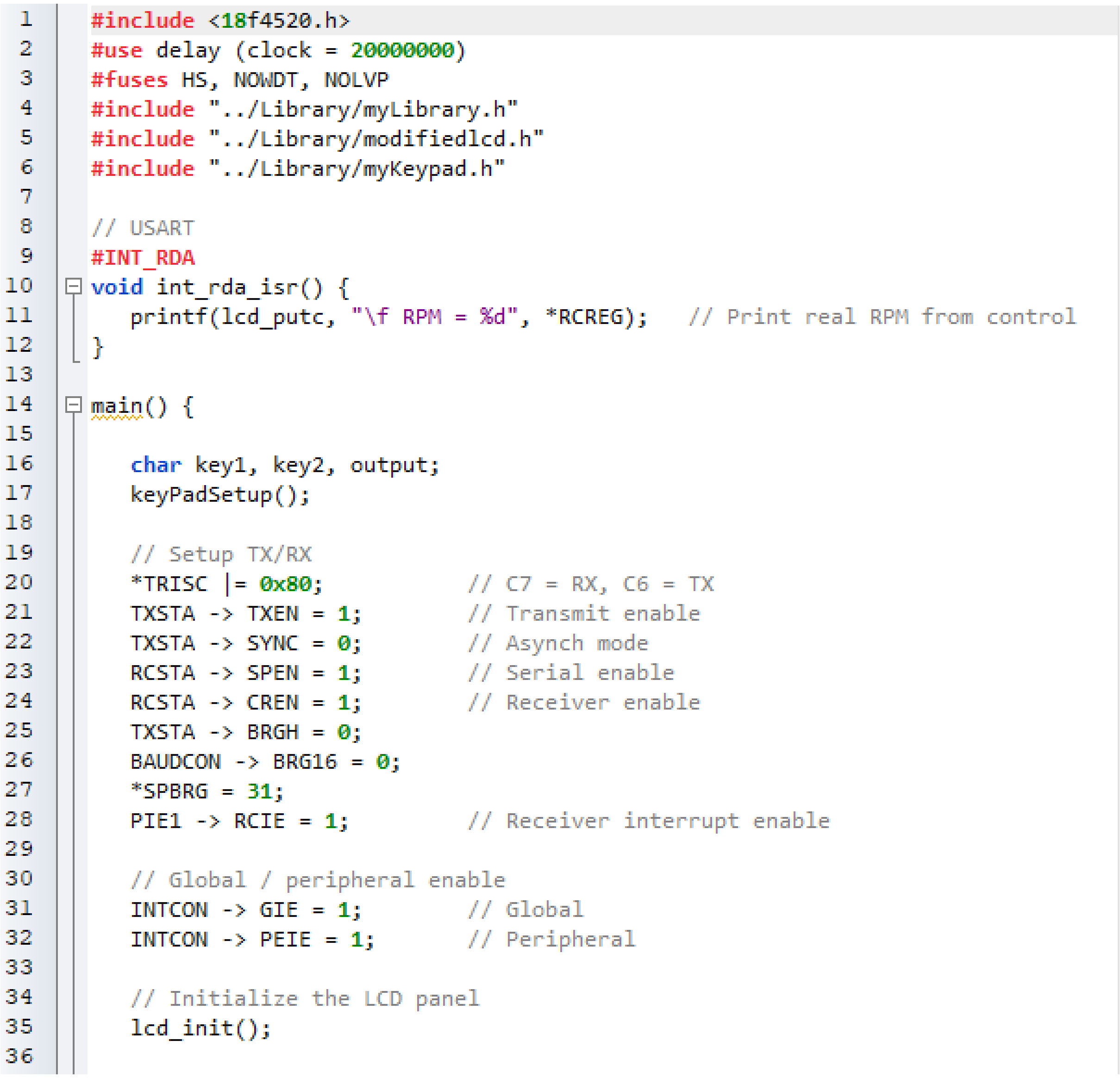


```
  // Setup timer1
  T1CON -> TMR1ON = 1;      // Timer1 is ON
  T1CON -> TMR1CS = 0;      // Fosc / 4
  T1CON -> T1CKPSx = 0;     // PS = 1;
  PIE1 -> TMR1IE = 1;       // Timer1 overflow interrupt is ON

  // Setup PWM
  CCP1CON -> CCPxMx = 0xC;
  *PR2 = 126;
  *CCPR1L = 10;
  T2CON -> TMR2ON = 1;

  // Setup stop button on B0
  ADCON1 -> PCFGx = 0x0F;   // Digital
  *TRISB = 0x01;            // Pin B0 as input
  INTCON2 -> INTEDG0 = 1;   // Rising edge
  INTCON -> INT0IE = 1;     // INT0 ON

  // Setup TX/RX
  TXSTA -> TXEN = 1;        // Transmit enable
  TXSTA -> SYNC = 0;        // Asynch mode
  RCSTA -> SPEN = 1;        // Serial enable
  RCSTA -> CREN = 1;        // Receiver enable
  TXSTA -> BRGH = 0;
  BAUDCON -> BRG16 = 0;
  *SPBRG = 31;
  PIE1 -> RCIE = 1;         // Receiver interrupt enable

  // Global / peripheral enable
  INTCON -> GIE = 1;        // Global
  INTCON -> PEIE = 1;       // Peripheral

  while( 1 ) {

    // Receive ERPM from monitor
    expectedRPM = *RCREG;

    // Set ERPM to PWM
    *CCPR1L = expectedRPM;

    // Calculate real RPM from capture
    realRPM = (int)(60.0 / ( 161.0 * (telapsed * t1c) ));

     // Send real RPM to monitor
    *TXREG = realRPM;
    delay_ms( 1000 );
```

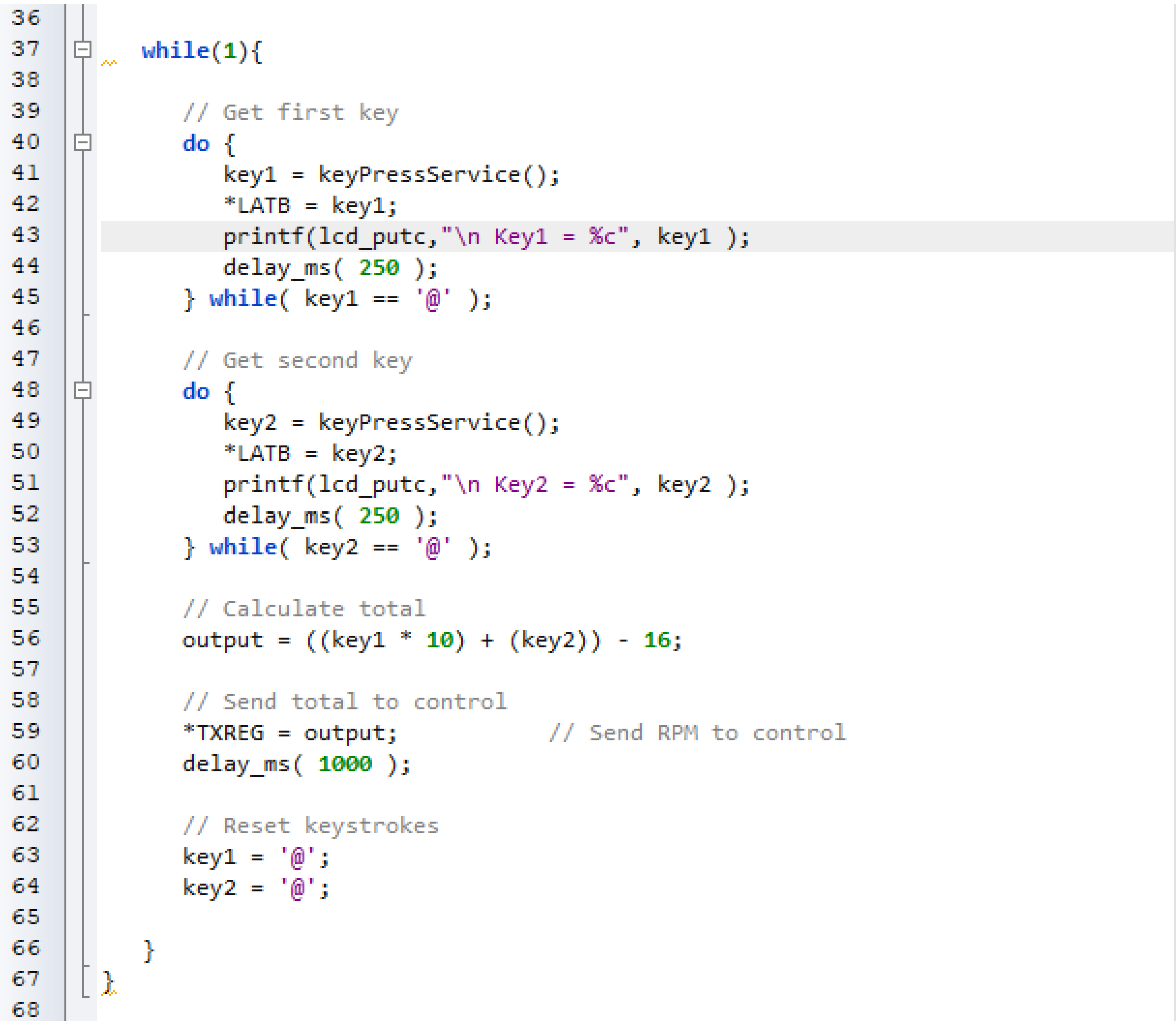


```

        // Control adjustments
        if ( realRPM > expectedRPM ) {
            *CCPR1L -= 1;
        }
        if ( expectedRPM > realRPM ) {
            *CCPR1L += 1;
        }
    }
}

```

MONITOR:



```
#include <18f4520.h>
#use delay (clock = 20000000)
#fuses HS, NOWDT, NOLVP
#include "../Library/myLibrary.h"
#include "../Library/modifiedlcd.h"
#include "../Library/myKeypad.h"

// USART
#INT_RDA
void int_rda_isr() {
   printf(lcd_putc, "\f RPM = %d", *RCREG);   // Print real RPM from control
}

main() {

   char key1, key2, output;
   keyPadSetup();

   // Setup TX/RX
   *TRISC |= 0x80;           // C7 = RX, C6 = TX
   TXSTA -> TXEN = 1;        // Transmit enable
   TXSTA -> SYNC = 0;        // Asynch mode
   RCSTA -> SPEN = 1;        // Serial enable
   RCSTA -> CREN = 1;        // Receiver enable
   TXSTA -> BRGH = 0;
   BAUDCON -> BRG16 = 0;
   *SPBRG = 31;
   PIE1 -> RCIE = 1;         // Receiver interrupt enable

   // Global / peripheral enable
   INTCON -> GIE = 1;        // Global
   INTCON -> PEIE = 1;       // Peripheral

   // Initialize the LCD panel
   lcd_init();

```



```
   while(1){

      // Get first key
      do {
         key1 = keyPressService();
         *LATB = key1;
         printf(lcd_putc,"\n Key1 = %c", key1 );
         delay_ms( 250 );
      } while( key1 == '@' );

      // Get second key
      do {
         key2 = keyPressService();
         *LATB = key2;
         printf(lcd_putc,"\n Key2 = %c", key2 );
         delay_ms( 250 );
      } while( key2 == '@' );

      // Calculate total
      output = ((key1 * 10) + (key2)) - 16;

      // Send total to control
      *TXREG = output;           // Send RPM to control
      delay_ms( 1000 );

      // Reset keystrokes
      key1 = '@';
      key2 = '@';

   }
}
```