James Martin

Madison College
Electrical Engineering and Electronics Technology
Microcontrollers

Project: Cruise Control

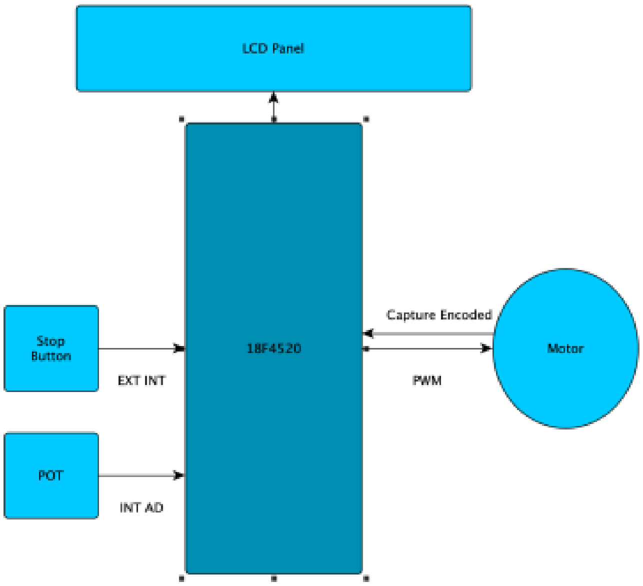
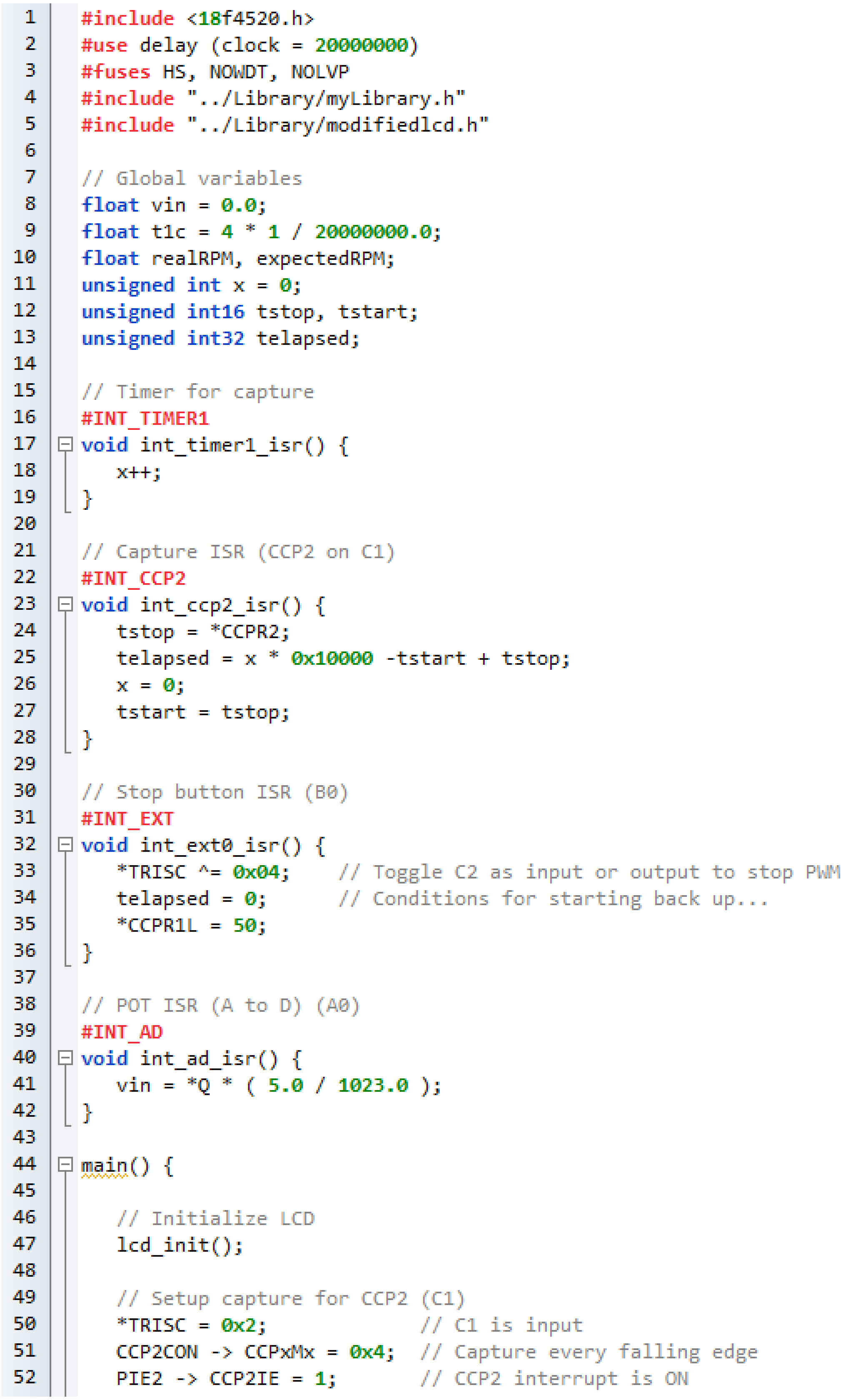
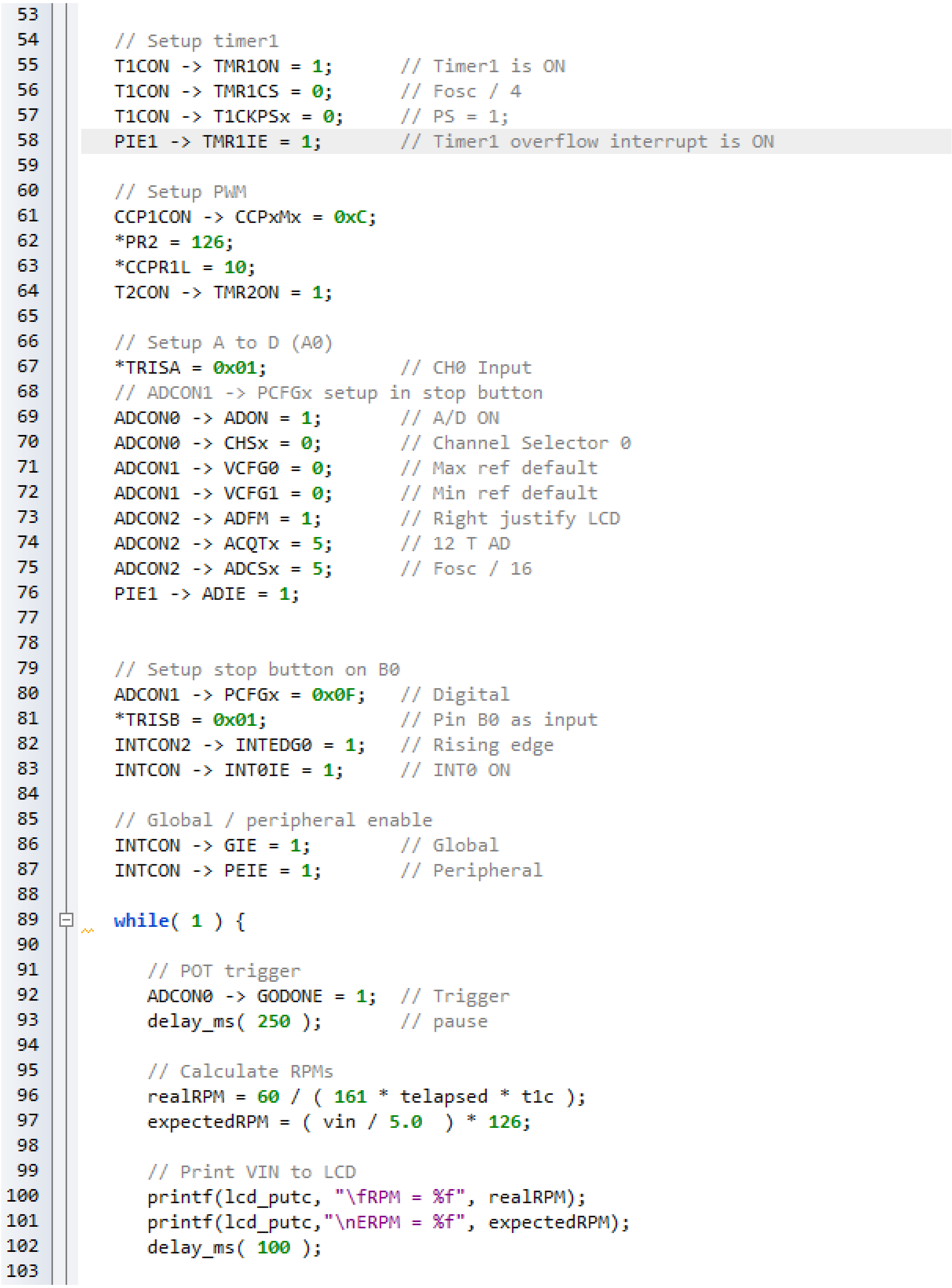


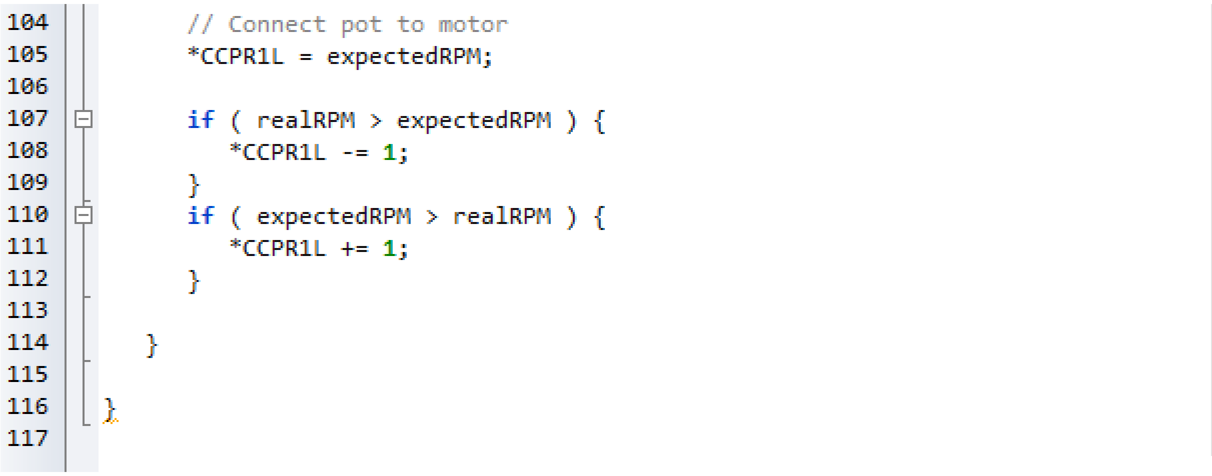
Figure 1: Control Block Diagram



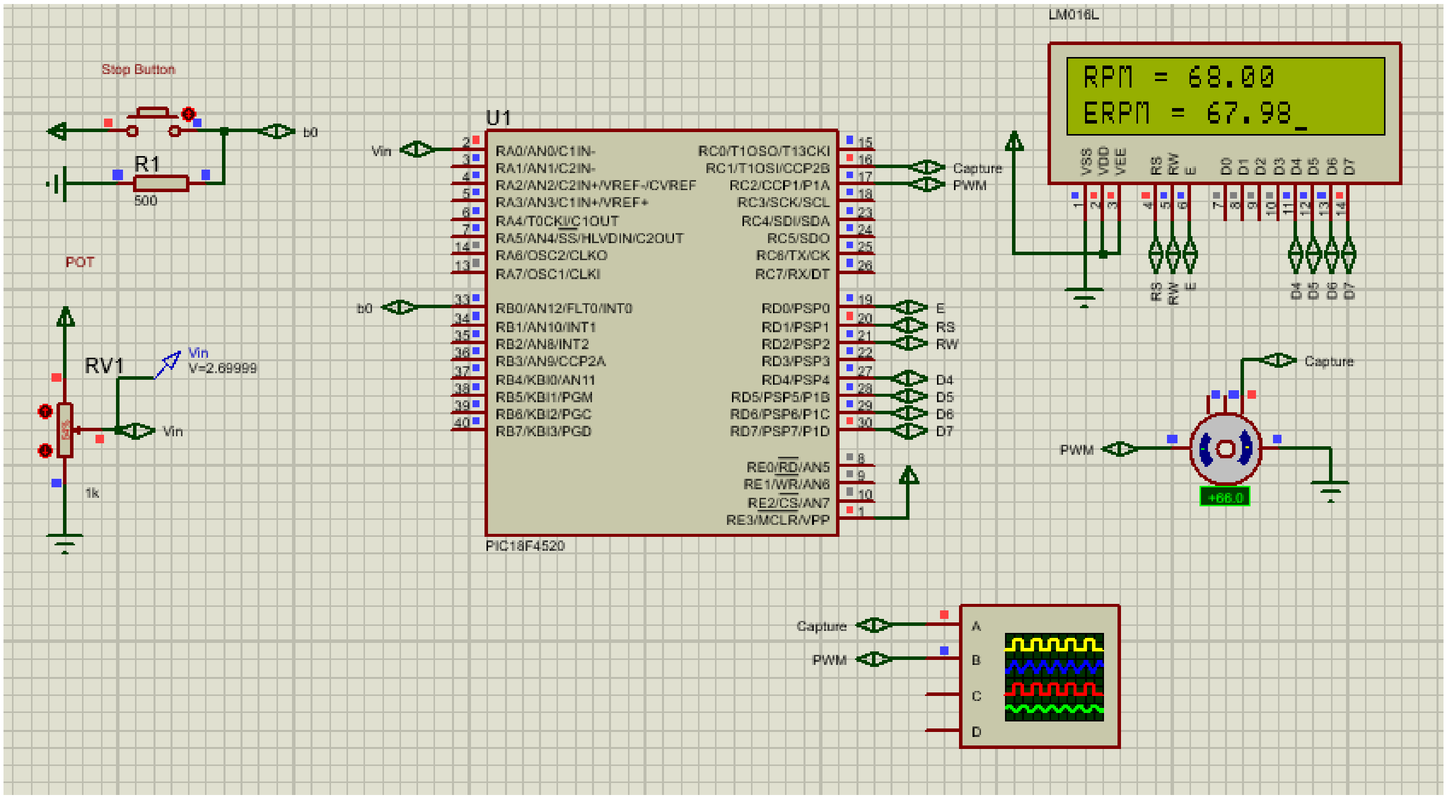
```c
#include <18f4520.h>
#use delay (clock = 20000000)
#fuses HS, NOWDT, NOLVP
#include "../Library/myLibrary.h"
#include "../Library/modifiedlcd.h"

// Global variables
float vin = 0.0;
float t1c = 4 * 1 / 20000000.0;
float realRPM, expectedRPM;
unsigned int 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;
   x = 0;
   tstart = tstop;
}

// 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;
}

// POT ISR (A to D) (A0)
#INT_AD
void int_ad_isr() {
   vin = *Q * ( 5.0 / 1023.0 );
}

main() {

   // Initialize LCD
   lcd_init();

   // Setup capture for CCP2 (C1)
   *TRISC = 0x2;             // C1 is input
   CCP2CON -> CCPxMx = 0x4;  // Capture every falling edge
   PIE2 -> CCP2IE = 1;       // CCP2 interrupt is ON
```



```c

    // 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 A to D (A0)
    *TRISA = 0x01;            // CH0 Input
    // ADCON1 -> PCFGx setup in stop button
    ADCON0 -> ADON = 1;       // A/D ON
    ADCON0 -> CHSx = 0;       // Channel Selector 0
    ADCON1 -> VCFG0 = 0;      // Max ref default
    ADCON1 -> VCFG1 = 0;      // Min ref default
    ADCON2 -> ADFM = 1;       // Right justify LCD
    ADCON2 -> ACQTx = 5;      // 12 T AD
    ADCON2 -> ADCSx = 5;      // Fosc / 16
    PIE1 -> ADIE = 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

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

    while( 1 ) {

       // POT trigger
       ADCON0 -> GODONE = 1;  // Trigger
       delay_ms( 250 );       // pause

       // Calculate RPMs
       realRPM = 60 / ( 161 * telapsed * t1c );
       expectedRPM = ( vin / 5.0  ) * 126;

       // Print VIN to LCD
       printf(lcd_putc, "\fRPM = %f", realRPM);
       printf(lcd_putc,"\nERPM = %f", expectedRPM);
       delay_ms( 100 );

```



```
      // Connect pot to motor
      *CCPR1L = expectedRPM;

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

   }

}
```


Stop Button
b0
R1
500
POT
RV1
Vin
V=2.69999
1k
U1
Vin
b0
RA0/AN0/C1IN-
RA1/AN1/C2IN-
RA2/AN2/C2IN+/VREF-/CVREF
RA3/AN3/C1IN+/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/HLVDIN/C2OUT
RA6/OSC2/CLKO
RA7/OSC1/CLKI
RB0/AN12/FLT0/INT0
RB1/AN10/INT1
RB2/AN8/INT2
RB3/AN9/CCP2A
RB4/KBI0/AN11
RB5/KBI1/PGM
RB6/KBI2/PGC
RB7/KBI3/PGD
RC0/T1OSO/T13CKI
RC1/T1OSI/CCP2B
RC2/CCP1/P1A
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5/P1B
RD6/PSP6/P1C
RD7/PSP7/P1D
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
RE3/MCLR/VPP
PIC18F4520
Capture
PWM
E
RS
RW
D4
D5
D6
D7
LM016L
RPM = 68.00
ERPM = 67.98_
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
RS RW E
D4 D5 D6 D7
Capture
PWM
+66.0
Capture
PWM
A
B
C
D