Microcontroller Code

Team 25, Fall 2012

1. Camera Triggering Module

```
#include <16F877A.h>
#device *=16      // Use 16-bit pointers, use 10-bit ADC

#fuses HS             // You may or may not want some of these ....
#fuses NOWDT
#fuses NOPROTECT
#fuses NOLVP
#fuses NODEBUG
#fuses NOPUT
#fuses NOBROWNOUT

#use delay(clock=20000000)
int16 pulse=0;   //forward pulse counter
int16 back=0;   //backward pulse counter
//int flag=0;

#INT_EXT
distance(){
  if(input(PIN_A1) && back==0){     //Speed Encoder CCW,car forward, not compensation mode
    pulse=pulse+1;          //LED blink every 316 pulses
    if(pulse==316 ){//&& flag==0){
      output_high(PIN_B4);
      delay_ms(1);
      output_low(PIN_B4);
      //flag=1;
      pulse=0;
    } //else if(pulse==500 && flag==1) {
     // output_low(PIN_B4);
     // flag=0;
     // pulse=0;
    //} else;
  } else if (input(PIN_A1) && back>0) {   //Speed Encoder CCW, car forward, compensation mode
      output_low(PIN_B4);
      back=back-1;
  } else {              //Speed Encoder CW, car backward, increase backward pulse counter
      output_low(PIN_B4);
      back=back+1;
```

```
  }
}
void main()
{

  setup_adc( ADC_OFF );
  setup_psp(PSP_DISABLED);
  setup_spi(SPI_SS_DISABLED);
  setup_timer_1 ( T1_INTERNAL | T1_DIV_BY_8 );
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);

  enable_interrupts(GLOBAL);
  enable_interrupts(INT_EXT);
  enable_interrupts(L_TO_H );
  while(TRUE){}
}
```

2. Motor Control (Build by previous group, modified)

```
#include <18F452.h>
#device adc=8
#FUSES NOWDT     //No Watch Dog Timer
#FUSES HS        //Highspeed Osc > 4mhz
#FUSES NOPUT       //Power Up Timer
#FUSES NOPROTECT  //Code not protected from reading
//#FUSES NODEBUG    //No Debug mode for ICD
#FUSES NOBROWNOUT //No brownout reset
#FUSES NOLVP      //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOCPD      //No EE protection
#use delay(clock=40000000)           // Sets crystal oscillator at 20 megahertz
#use i2c(Master, sda=PIN_C4, scl=PIN_C3)
#use rs232(baud=9600, rcv=PIN_C7, stream=READ)

//main program
void main(){
  char prev;
  char curr;
  int i;
  char cbuff[4];
  delay_ms(100);
  while(true){
    for(i=0; i<4; i++){
```

```
    cbuff[i]=fgetc(READ);
}
prev = cbuff[2];
if(input(PIN_B7)){
    output_low(PIN_B1);
    output_low(PIN_B2);
    output_low(PIN_B3);
}
else if(curr != prev){       //curr != prev
  if(prev == 'F'){
    output_low(PIN_B1);
    output_low(PIN_B2);
    output_low(PIN_B3);
    curr = prev;
  }
  else if(prev == '0'){
    output_low(PIN_B1);
    output_low(PIN_B2);
    output_low(PIN_B3);
    curr = prev;
  }
  else if(prev == '1'){
    output_low(PIN_B1);
    output_high(PIN_B2);
    output_low(PIN_B3);
    curr = prev;
  }
  else if(prev == '2'){
    output_high(PIN_B1);
    output_low(PIN_B2);
    output_high(PIN_B3);
    curr = prev;
  }
  else if(prev == '3'){
    output_low(PIN_B1);
    output_high(PIN_B2);
    output_high(PIN_B3);
    curr = prev;
  }
  else if(prev == '4'){
    output_high(PIN_B1);
    output_high(PIN_B2);
```

```
          output_high(PIN_B3);
          curr = prev;
        }

      }
      delay_ms(1);
    }
  }
```

3. Wireless Speed Display-receiver

```
#include <16F877A.h>
#device *=16      // Use 16-bit pointers, use 10-bit ADC

#fuses HS             // You may or may not want some of these ....
#fuses NOWDT
#fuses NOPROTECT
#fuses NOLVP
#fuses NODEBUG
#fuses NOPUT
#fuses NOBROWNOUT

#use delay(clock=20000000)
#use rs232(baud=9600, xmit=PIN_C6, stream=PC)
#use rs232(baud=9600, rcv=PIN_C7, stream=READ)

void main()
{
  char letter[4];
  int i;
  setup_adc( ADC_OFF );
  setup_psp(PSP_DISABLED);
  setup_spi(SPI_SS_DISABLED);
  setup_timer_1 ( T1_INTERNAL | T1_DIV_BY_8 );
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);
  while(TRUE) {
      for(i=0; i<4; i++){
        letter[i]=fgetc(READ);
      }
      if(letter[2]=='0'){
        output_low(PIN_A0);
        output_low(PIN_A1);
```

```
  output_low(PIN_A2);
  output_low(PIN_A3);

  output_high(PIN_B0);
  output_high(PIN_B1);
  output_high(PIN_B2);
  output_high(PIN_B3);
  output_high(PIN_B4);
  output_high(PIN_B5);
  output_low(PIN_B6);
}
else if(letter[2]=='1'){
  output_high(PIN_A0);
  output_low(PIN_A1);
  output_low(PIN_A2);
  output_low(PIN_A3);

  output_low(PIN_B0);
  output_high(PIN_B1);
  output_high(PIN_B2);
  output_low(PIN_B3);
  output_low(PIN_B4);
  output_low(PIN_B5);
  output_low(PIN_B6);
}
else if(letter[2]=='2'){
  output_low(PIN_A0);
  output_high(PIN_A1);
  output_low(PIN_A2);
  output_low(PIN_A3);

  output_high(PIN_B0);
  output_high(PIN_B1);
  output_low(PIN_B2);
  output_high(PIN_B3);
  output_high(PIN_B4);
  output_low(PIN_B5);
  output_high(PIN_B6);
}
else if(letter[2]=='3'){
  output_high(PIN_A0);
  output_high(PIN_A1);
```

```
  output_low(PIN_A2);
  output_low(PIN_A3);

  output_high(PIN_B0);
  output_high(PIN_B1);
  output_high(PIN_B2);
  output_high(PIN_B3);
  output_low(PIN_B4);
  output_low(PIN_B5);
  output_high(PIN_B6);
}
else if(letter[2]=='4'){
  output_low(PIN_A0);
  output_low(PIN_A1);
  output_high(PIN_A2);
  output_low(PIN_A3);

  output_low(PIN_B0);
  output_high(PIN_B1);
  output_high(PIN_B2);
  output_low(PIN_B3);
  output_low(PIN_B4);
  output_high(PIN_B5);
  output_high(PIN_B6);
}
else if(letter[2]=='5'){
  output_high(PIN_A0);
  output_low(PIN_A1);
  output_high(PIN_A2);
  output_low(PIN_A3);

  output_high(PIN_B0);
  output_low(PIN_B1);
  output_high(PIN_B2);
  output_high(PIN_B3);
  output_low(PIN_B4);
  output_high(PIN_B5);
  output_high(PIN_B6);
}
else {
  output_low(PIN_A0);
  output_low(PIN_A1);
```

```
        output_low(PIN_A2);
        output_low(PIN_A3);

        output_high(PIN_B0);
        output_high(PIN_B1);
        output_high(PIN_B2);
        output_high(PIN_B3);
        output_high(PIN_B4);
        output_high(PIN_B5);
        output_high(PIN_B6);
      }
  }
}
```

4. Wireless Speed Display-transmitter

```
#include <16F877A.h>
#device *=16      // Use 16-bit pointers, use 10-bit ADC

#fuses HS             // You may or may not want some of these ....
#fuses NOWDT
#fuses NOPROTECT
#fuses NOLVP
#fuses NODEBUG
#fuses NOPUT
#fuses NOBROWNOUT

#use delay(clock=20000000)
#use rs232(baud=9600, xmit=PIN_C6, stream=PC)
#use rs232(baud=9600, rcv=PIN_C7, stream=READ)

int16 pulse=0;   //pulse counter
int16 flag=0;    //timer count

#INT_TIMER1
speed(){
  flag=flag+1;
  if(flag>=10){
  flag=0;
  //freq=(double)pulse/(double)(10*0.1048576);
  set_timer1(0);
  if(pulse<245){
    fprintf(PC,"0000\n");
```

```
    output_low(PIN_B4);
    output_high(PIN_B3);
  }
  else if (pulse>=245 && pulse<735){
    fprintf(PC,"1111\n");
    output_high(PIN_B4);
    output_low(PIN_B3);
  }
  else if(pulse>=735 && pulse<1233){
    fprintf(PC,"2222\n");
    output_low(PIN_B4);
    output_low(PIN_B3);
  }
  else if(pulse>=1233 && pulse<1775){
    fprintf(PC,"3333\n");
    output_high(PIN_B4);
    output_low(PIN_B3);
  }
  else if(pulse>=1775 && pulse<2205){
    fprintf(PC,"4444\n");
    output_low(PIN_B4);
    output_low(PIN_B3);
  }
  else{
    fprintf(PC,"5555\n");
    output_low(PIN_B3);
  }
  pulse = 0;
  }
}

#INT_EXT
count(){
  pulse=pulse+1;
}

void main()
{

  setup_adc( ADC_OFF );
  setup_psp(PSP_DISABLED);
  setup_spi(SPI_SS_DISABLED);
```

```
  setup_timer_1 ( T1_INTERNAL | T1_DIV_BY_8 );
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);

  enable_interrupts(GLOBAL);
  enable_interrupts(INT_TIMER1);
  enable_interrupts(INT_EXT);
  enable_interrupts(L_TO_H );

  setup_timer_1(T1_INTERNAL | T1_DIV_BY_8);
  set_timer1(0);
  while(TRUE){}
}
```

5. Obstacle Detection Module

```
#include <16F877A.h>    // <-- This is the type of PIC you're using.
                        //    Other supported PICs are in:
                        //    C:\Program Files\PICC\Devices

//#device ICD=TRUE
#device *=16 ADC=10     // Use 16-bit pointers, use 10-bit ADC

#fuses HS               // You may or may not want some of these ....
#fuses NOWDT
#fuses NOPROTECT
#fuses NOLVP
#fuses NODEBUG
#fuses NOPUT
#fuses NOBROWNOUT

#use delay(clock=20000000)
void main()
{
  int16 voltage=0;
  int flag=0;
  setup_adc_ports(AN0_AN1_AN3);
  setup_adc(ADC_CLOCK_DIV_64);
  setup_psp(PSP_DISABLED);
  setup_spi(SPI_SS_DISABLED);
  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
```

```
  setup_comparator(NC_NC_NC_NC);
  setup_vref(FALSE);


  set_adc_channel(0);  // selects pin A0 to read from
  while(TRUE){
  output_high(PIN_B2);
    voltage=read_adc();
    if(voltage<90 || (flag==1 && voltage<150 )) {
      output_high(PIN_B4);
      flag=1;
    } else if(flag==1 && voltage>=150 ){
      output_low(PIN_B4);
      flag=0;
    } else
      output_low(PIN_B4);
  }
}
```