

```
#include <stdio.h>
#define _LEGACY_HEADERS
#include <htc.h>

__CONFIG( EXTCLK & HS & WDTDIS & PWRTEN & MCLRDIS & UNPROTECT & BORDIS & IESODIS & FCMEN );

void InitSSP(void);
void InitPWM(void);
void ZombieConvert(void);
void HumanMode(void);
void setPWM(void);

#define DATA_LENGTH 3

static unsigned char SSP_DataLength = 0;
static signed char SpeedData = 0;
static signed char DirnData = 0;
static signed char FlagData = 0;
static unsigned char DataReceived = 0;
static signed char SSP_DataArray[DATA_LENGTH+1];
static unsigned char DirnPins = 0;

void main(void)
{
    InitSSP();
    InitPWM();
    while(1)
    {
        if(DataReceived)
        {
            SpeedData = SSP_DataArray[0];
            DirnData = SSP_DataArray[1];
            FlagData = SSP_DataArray[2];

            if(FlagData & 0b00100000) //If zombie
            {
                ZombieConvert();
                HumanMode();
            }
            else {HumanMode();}

            setPWM();
            DataReceived = 0;
        }
    }
}

void InitSSP(void)
{
    ANSEL = 0;
    ANSELH = 0;
```

```

TRISA2 = 0;
TRISC0 = 0;    // Direction and Brake pins for Motors
TRISC1 = 0;
TRISC2 = 0;

CKP = 1;    //SSP mode select
SSPM2 = 1;    //slave mode with slave select enabled

TRISC6 = 1;    //output slave select line RC6 to digital, input

TRISB4 = 1;    //set RB4/SDI to digital input
TRISC7 = 0;    //set RC7/SDO to digital output
TRISB6 = 1;    //set RB6/SCK to digital input

SSPEN = 1;    //SSP enabled

GIE = 1;
PEIE = 1;
SSPIE = 1;    //Enable Interrupts

```

```

}

```

```

void interrupt ISR(void)

```

```

{
    if(SSPIF)    //If Data Received
    {
        SSP_DataArray[SSP_DataLength++] = SSPBUF;    //Store Data
        if(SSP_DataLength == DATA_LENGTH)    //If Data Length as required
        {
            SSP_DataLength = 0;    //Reset
            DataReceived = 1;    //Set DataReceived Flag High
        }
        SSPIF = 0;    //Clear Flag
    }
}

```

```

void InitPWM(void)

```

```

{
    TRISC5 = 0;    //set CCP1 pin, i.e PWM pin as output
    PR2 = 250;    //20khz period
    TMR2ON = 1;    //Timer 2 ON
    CCPR1L = 0;    //Initialize PWM to 0,
    CCP1CON = 0b00001100;    //Initialize Timer 2
    RA2 = 0;
    RC0 = 1;
    RC1 = 0;
    RC2 = 1;    //Initalize Brakes as high
}

```

```

void ZombieConvert()

```

```

{
    if(FlagData & 0b00000001) SpeedData = (SpeedData/4)*3;    //If speed enhanced
    else {SpeedData = SpeedData/2;}
}

```

```

if(FlagData & 0b00010000) SpeedData = SpeedData*(-1);           //If F/B swapped

if(FlagData & 0b00001000) DirnData = DirnData*(-1);           //If R/L swapped
}

void HumanMode()
{
  if(SpeedData < 0)
  {
    SpeedData = SpeedData*-1;
    if( (DirnData <=-40) && (FlagData & 0b00000010) ) DirnData = DirnData/2;           //
    if turning left and Left turn degraded           CCW
    else if ( (DirnData >40) && (FlagData & 0b00000100) ) DirnData = DirnData/2;           //
    else if turning right and right turn degraded CW

    if((DirnData <=40) && (DirnData>=-40)) DirnPins = 0b00001010;           //
    go straight

    if((DirnData <=-40) && (DirnData>=-100)) DirnPins = 0b00001110;           //
    turn left (CCW), with just one propeller on
    else if ((DirnData <=100) && (DirnData>40)) DirnPins = 0b00001011;           //
    turn right (CW), with just one propeller on

    if (DirnData <-100) DirnPins = 0b00000010;           // turn CCW, with
    propellers in opposite dirn
    else if ( DirnData>100 ) DirnPins = 0b00001000;           // turn CW, with
    propellers in opposite dirn
  }
  else
  {
    if( (DirnData <=-40) && (FlagData & 0b00000010) ) DirnData = DirnData/2;           //if
    turning left while moving fwd and left turn degraded           CW
    else if (( DirnData >40) && (FlagData & 0b00000100) ) DirnData = DirnData/2;
    //else if turning right while moving back and right turn degraded           CCW

    if((DirnData <=40) && (DirnData>=-40)) DirnPins = 0b00000000;           //
    go straight

    if((DirnData <=-40) && (DirnData>=-100)) DirnPins = 0b00000100;           //
    turn left (CW), with just one propeller on
    else if ((DirnData <=100) && (DirnData>40)) DirnPins = 0b00000001;           //
    turn right (CCW), with just one propeller on

    if (DirnData <-100) DirnPins = 0b00001000;           // turn CW, with
    propellers in opposite dirn
    else if (DirnData>100) DirnPins = 0b00000010;           // turn CCW, with
    propellers in opposite dirn
  }
}

void setPWM(void)
{

```

```
if(SpeedData>=0 && SpeedData <=20)
{
    CCPR1L = 0;           //PWM = 0;
}
else if (SpeedData >20 && SpeedData<= 40)
{
    CCPR1L = 125;
} //set PWM 50 percent
else if (SpeedData >40 && SpeedData<=60)
{
    CCPR1L = 175;
} //set PWM 70 percent
else if (SpeedData >60 && SpeedData<=80)
{
    CCPR1L = 200;
} //set PWM 80 percent
else if (SpeedData >80 && SpeedData<=100)
{
    CCPR1L = 225;
} //set PWM 90 percent
else
{
    CCPR1L = 251;
} //set PWM 100 percent
RA2 = (DirnPins & 0b00001000)!=0;
RC0 = (DirnPins & 0b00000100)!=0;
RC1 = (DirnPins & 0b00000010)!=0;
RC2 = (DirnPins & 0b00000001)!=0; // set dirn pins to DirnData
}
```