

```

/* Code for MECH307 two wheeled balancing robot based on:
YourDuinoStarter Example: nRF24L01 Transmit Joystick values

- WHAT IT DOES: Reads Analog values on A0, A1 and transmits
them over a nRF24L01 Radio Link to another transceiver.
- SEE the comments after "/" on each line below
- CONNECTIONS: nRF24L01 Modules See:
http://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo
  1 - GND
  2 - VCC 3.3V !!! NOT 5V
  3 - CE to Arduino pin 8
  4 - CSN to Arduino pin 9
  5 - SCK to Arduino pin 13
  6 - MOSI to Arduino pin 11
  7 - MISO to Arduino pin 12
  8 - UNUSED
-
Analog Joystick or two 10K potentiometers:
GND to Arduino GND
VCC to Arduino +5V
X Pot to Arduino A0
Y Pot to Arduino A1

- V1.00 11/26/13
Based on examples at http://www.bajdi.com/
Questions: terry@yourduino.com */

/*-----( Import needed libraries )-----*/
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <stdint.h>
/*-----( Declare Constants and Pin Numbers )-----*/
#define CE_PIN 8
#define CSN_PIN 9
#define JOYSTICK_X A0
#define JOYSTICK_Y A1
#define JOYSTICK_BUTTON 7
#define BUTTON1 6
#define BUTTON2 5
#define PWR_LED 3

// NOTE: the "LL" at the end of the constant is "LongLong" type
const uint64_t pipe = 0xE8E8F0F0E1LL; // Define the transmit pipe

int lasttime; //used for timing of LED fading
signed int LEDVal = 0; //used for keeping track of LED PWM duty cycle
bool countDir = 1; //used for keeping track of whether counting up or down
bool lastB2state = 0;

/*-----( Declare objects )-----*/

```

```

RF24 radio(CE_PIN, CSN_PIN); // Create a Radio
/*-----( Declare Variables )-----*/
uint8_t inputArray[32]; // 5 element array holding Joystick and button readings

void setup() //***** SETUP: RUNS ONCE *****/
{
  // Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(pipe);

  pinMode(JOYSTICK_BUTTON, INPUT);
  pinMode(BUTTON1, INPUT);
  pinMode(BUTTON2, INPUT);
  pinMode(PWR_LED, OUTPUT);

  digitalWrite(PWR_LED, LOW);
  lasttime = millis();
  initializeArrayZeros(inputArray);
} //--(end setup )---

void loop() //***** LOOP: RUNS CONSTANTLY *****/
{
  inputArray[0] = analogRead(JOYSTICK_X)/4;
  inputArray[1] = 255-analogRead(JOYSTICK_Y)/4; //255-y to invert
  inputArray[2] = digitalRead(BUTTON1);

  if(lastB2state == 1){
    inputArray[3] = 0;
  }
  else{
    lastB2state = digitalRead(BUTTON2);
    inputArray[3] = lastB2state;
  }

  inputArray[4] = digitalRead(JOYSTICK_BUTTON);

  if(inputArray[0] == 0){ //don't send a 0 in this byte. Used to determine whether remote is sending data.
    inputArray[0] = 1;
  }

  radio.write( inputArray, sizeof(inputArray) );
  //Debug Lines to print data
  // Serial.print(inputArray[0]); Serial.print(" ");
  // Serial.print(inputArray[1]); Serial.print(" ");
  // Serial.print(inputArray[2]); Serial.print(" ");
  // Serial.print(inputArray[3]); Serial.print(" ");
  // Serial.print(inputArray[4]); Serial.print(" ");
  // Serial.print(inputArray[5]); Serial.print(" ");
  // Serial.print(inputArray[6]); Serial.print(" ");
  // Serial.print(inputArray[7]); Serial.print(" ");

```

```

// Serial.println(inputArray[8]);

UpdateStatusLED();

} // --(end main loop) ---

/*-----( Declare User-written Functions )-----*/
void initializeArrayZeros(uint8_t* array){
  for(int i=0; i<sizeof(array); i++){
    array[i] = 0;
  }
  return;
}

void UpdateStatusLED(){
  if(millis()-lasttime > 10){
    lasttime = millis();
    if(countDir == 1){ //counting up?
      if(LEDVal < 255){
        LEDVal += 4;
      }
    }

    if(countDir == 0){ //counting down?
      if(LEDVal > 0){
        LEDVal -= 4;
      }
    }

    if(LEDVal > 255){ //check high limit
      LEDVal = 255;
      countDir = 0;
    }
    if(LEDVal < 0){ //check low limit
      LEDVal = 0;
      countDir = 1;
    }
    analogWrite(PWR_LED, LEDVal);
  }
  return;
} //END UpdateStatusLED()

//NONE
//***** ( THE END )*****

```