```
/*A basic 4 channel transmitter using the nRF24L01
module.*/

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

/*Create a unique pipe out. The receiver has to
  wear the same unique code*/

const uint64_t pipeOut = 0xE8E8F0F0E1LL;
//IMPORTANT: The same as in the receiver

RF24 radio(9, 10); // select  CSN  pin

// The sizeof this struct should not exceed 32 bytes
// This gives us up to 32 8 bits channals
struct MyData {
  byte throttle;
  byte yaw;
  byte pitch;
  byte roll;
  byte AUX1;
  byte AUX2;
};

MyData data;
```

```cpp
void resetData()
{
  //This are the start values of each channal
  // Throttle is 0 in order to stop the motors
  //127 is the middle value of the 10ADC.

  data.throttle = 0;
  data.yaw = 127;
  data.pitch = 127;
  data.roll = 127;
  data.AUX1 = 0;
  data.AUX2 = 0;
}

void setup()
{
  //Start everything up
  radio.begin();
  radio.setAutoAck(false);
  radio.setDataRate(RF24_250KBPS);
  radio.openWritingPipe(pipeOut);
  resetData();
}

/********************************************************/

// Returns a corrected value for a joystick
position that takes into account
```

```cpp
// the values of the outer extents and the middle
of the joystick range.
int mapJoystickValues(int val, int lower, int
middle, int upper, bool reverse)
{
  val = constrain(val, lower, upper);
  if ( val < middle )
    val = map(val, lower, middle, 0, 128);
  else
    val = map(val, middle, upper, 128, 255);
  return ( reverse ? 255 - val : val );
}

void loop()
{
  // The calibration numbers used here should be
measured
  // for your joysticks till they send the correct
values.
  data.throttle = mapJoystickValues(
analogRead(A4), 0, 521, 1023, true ); //Change to
A4 for potentiometer controlled
  data.yaw      = mapJoystickValues(
analogRead(A1), 0, 535, 1023, false ); //525
  data.pitch    = mapJoystickValues(
analogRead(A2), 0, 501, 1023, true ); //511
  data.roll     = mapJoystickValues(
analogRead(A3), 0, 512, 1023, false ); //502
```

```
    data.AUX1         = digitalRead(2); //The 2 toggle
switches
    data.AUX2         = digitalRead(3);

    radio.write(&data, sizeof(MyData));
}
```