#include <Psx.h>

#define dataPin 2

#define cmndPin 3

#define attPin 4

#define clockPin 5

#define LEDPin 12

#define IP\_1 8

#define IP\_2 9

#define IP\_3 10

#define IP\_4 11

Psx Psx;

unsigned int data = 0;

void setup()

{

Psx.setupPins(dataPin, cmndPin, attPin, clockPin, 10);

pinMode(LEDPin, OUTPUT);

Serial.begin(9600);

pinMode(IP\_1,OUTPUT);

pinMode(IP\_2,OUTPUT);

pinMode(IP\_3,OUTPUT);

pinMode(IP\_4,OUTPUT);

}

void loop()

{

data = Psx.read();

switch(data)

{

case psxR2 : digitalWrite(LEDPin, HIGH);

Serial.println("R2");

high();

break;

case psxR1 : digitalWrite(LEDPin, HIGH);

Serial.println("R1");

high();

break;

case psxL2 : digitalWrite(LEDPin, HIGH);

Serial.println("L2");

high();

break;

case psxL1 : digitalWrite(LEDPin, HIGH);

Serial.println("L1");

high();

break;

case psxUp : digitalWrite(LEDPin, HIGH);

Serial.println("Up");

forward();

break;

case psxDown: digitalWrite(LEDPin, HIGH);

Serial.println("Down");

backward();

break;

case psxRight:digitalWrite(LEDPin, HIGH);

Serial.println("Right");

right();

break;

case psxLeft :digitalWrite(LEDPin, HIGH);

Serial.println("Left");

left();

break;

case psxTri : digitalWrite(LEDPin, HIGH);

Serial.println("Triangle");

low();

break;

case psxSqu : digitalWrite(LEDPin, HIGH);

Serial.println("Square");

low();

break;

case psxO : digitalWrite(LEDPin, HIGH);

Serial.println("Circle");

low();

break;

case psxX : digitalWrite(LEDPin, HIGH);

Serial.println("Cross");

low();

break;

case psxSlct :digitalWrite(LEDPin, HIGH);

Serial.println("Select");

low();

break;

case psxStrt :digitalWrite(LEDPin, HIGH);

Serial.println("Start");

low();

break;

default : digitalWrite(LEDPin,LOW);

}

}

void forward()

{

digitalWrite(IP\_1,HIGH);

digitalWrite(IP\_2,LOW);

digitalWrite(IP\_3,LOW);

digitalWrite(IP\_4,HIGH);

}

void backward()

{

digitalWrite(IP\_1,LOW);

digitalWrite(IP\_2,HIGH);

digitalWrite(IP\_3,HIGH);

digitalWrite(IP\_4,LOW);

}

void right()

{

digitalWrite(IP\_1,HIGH);

digitalWrite(IP\_2,LOW);

digitalWrite(IP\_3,HIGH);

digitalWrite(IP\_4,LOW);

}

void left()

{

digitalWrite(IP\_1,LOW);

digitalWrite(IP\_2,HIGH);

digitalWrite(IP\_3,LOW);

digitalWrite(IP\_4,HIGH);

}

void low()

{

digitalWrite(IP\_1,LOW);

digitalWrite(IP\_2,LOW);

digitalWrite(IP\_3,LOW);

digitalWrite(IP\_4,LOW);

}

void high()

{

digitalWrite(IP\_1,HIGH);

digitalWrite(IP\_2,HIGH);

digitalWrite(IP\_3,HIGH);

digitalWrite(IP\_4,HIGH);

}