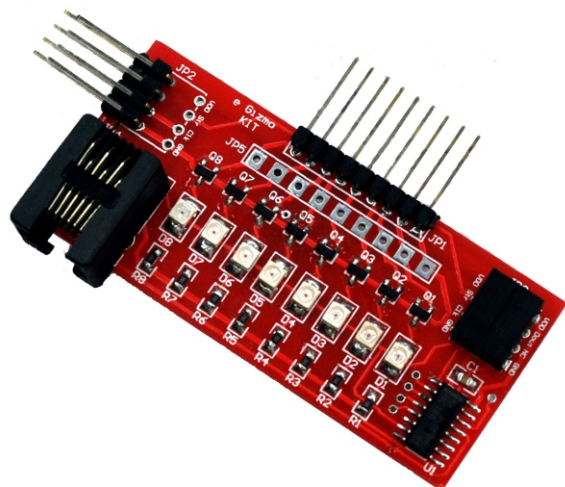


Input Expander Module

Hardware Manual Rev 1r0



Features & Specifications

The input expander module extends 3 microcontroller I/O ports along with the output expander module by adding 8 I/O ports. Individually, it acts as a binary counter indicated by the number of LED indicators lit. Works with 3.3V and 5V logic.

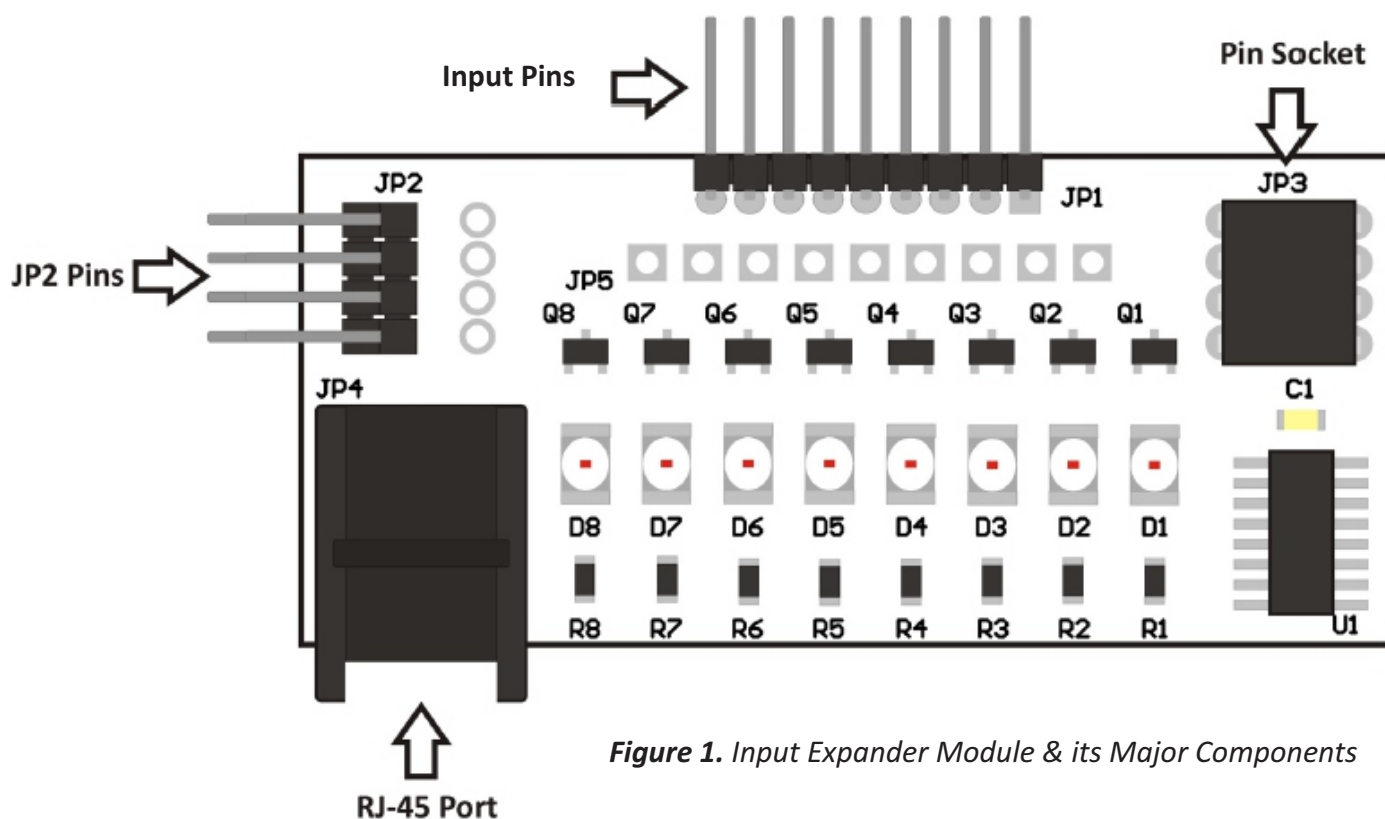


Figure 1. Input Expander Module & its Major Components

PIN ASSIGNMENTS

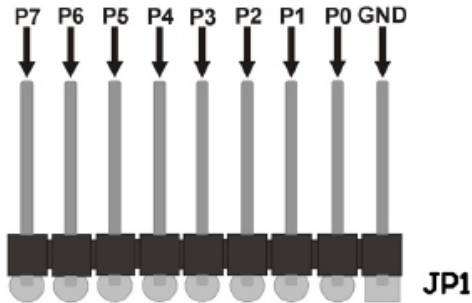


Figure 2. JP1 Pin I.D. Illustration

Table 2. JP1 Pin I.D. & Descriptions

Pin I.D.	Descriptions
P7	Input
P6	Input
P5	Input
P4	Input
P3	Input
P2	Input
P1	Input
P0	Input
GND	Ground

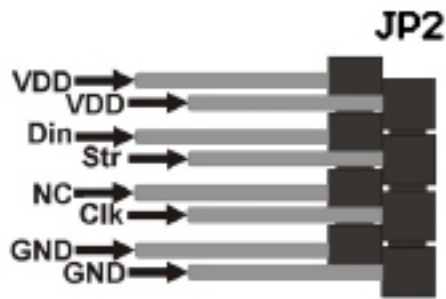


Figure 3. JP2 Pin I.D. Illustration

Table 3. JP2 Pin I.D. & Descriptions

Pin I.D.	Descriptions
VDD	Supply voltage +5V
Str	Strobe
Clk	Clock(Hi,Lo)
GND	GROUND
VDD	Supply voltage +5V
Din	Data In
NC	No Change
GND	Ground

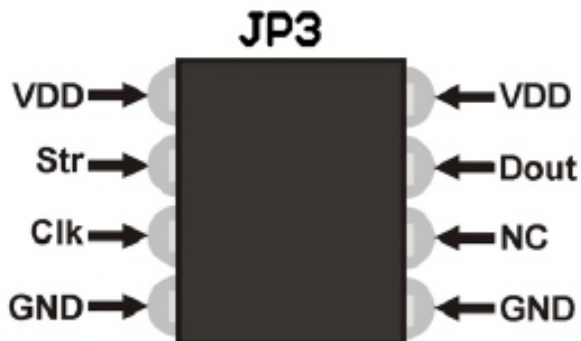


Figure 4. JP3 for Cascade, Pin Slot Pin I.D. Illustration

Table 3. JP3 Pin I.D. & Descriptions

Pin I.D.	Descriptions
VDD	Supply voltage +5V
Str	Strobe
Clk	Clock(Hi,Lo)
GND	GROUND
VDD	Supply voltage +5V
Din	Data Out
NC	No Change
GND	Ground

PCB BOARD PRESENTATION

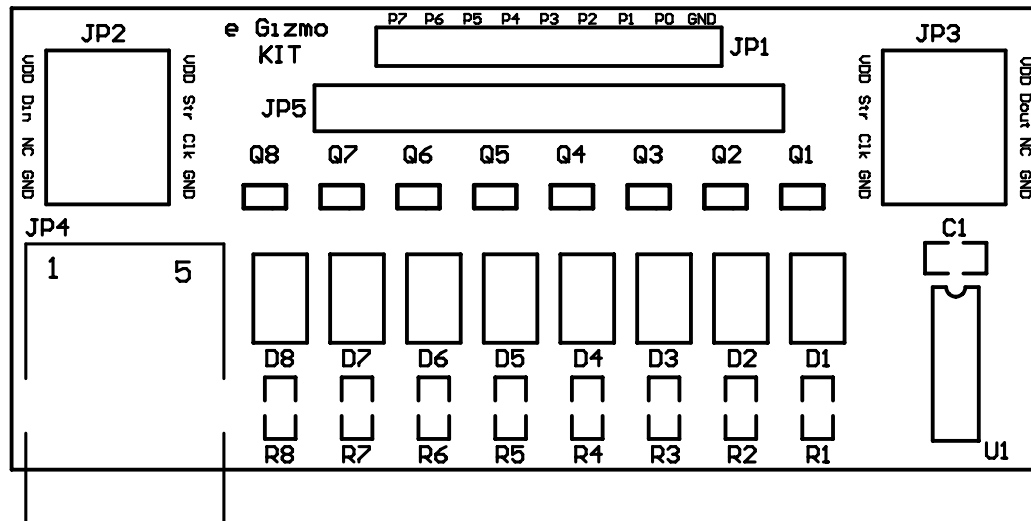


Figure 5. Input Expander Module (silkscreen layout)

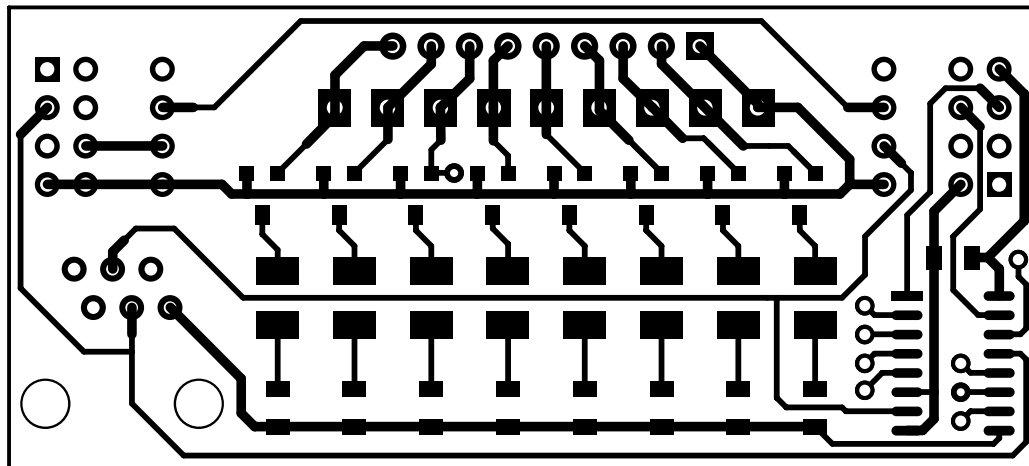


Figure 6. Input Expander Module PCB Copper Pattern (Top Layer)

PCB BOARD PRESENTATION

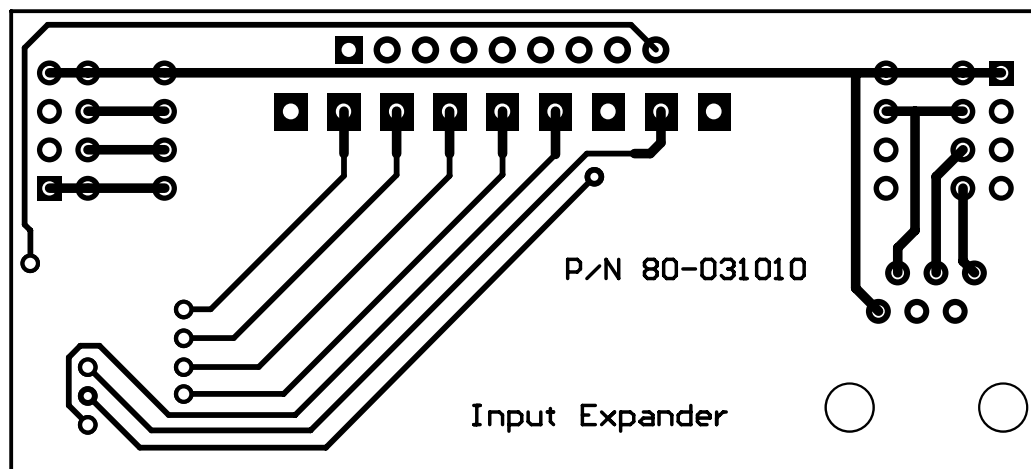
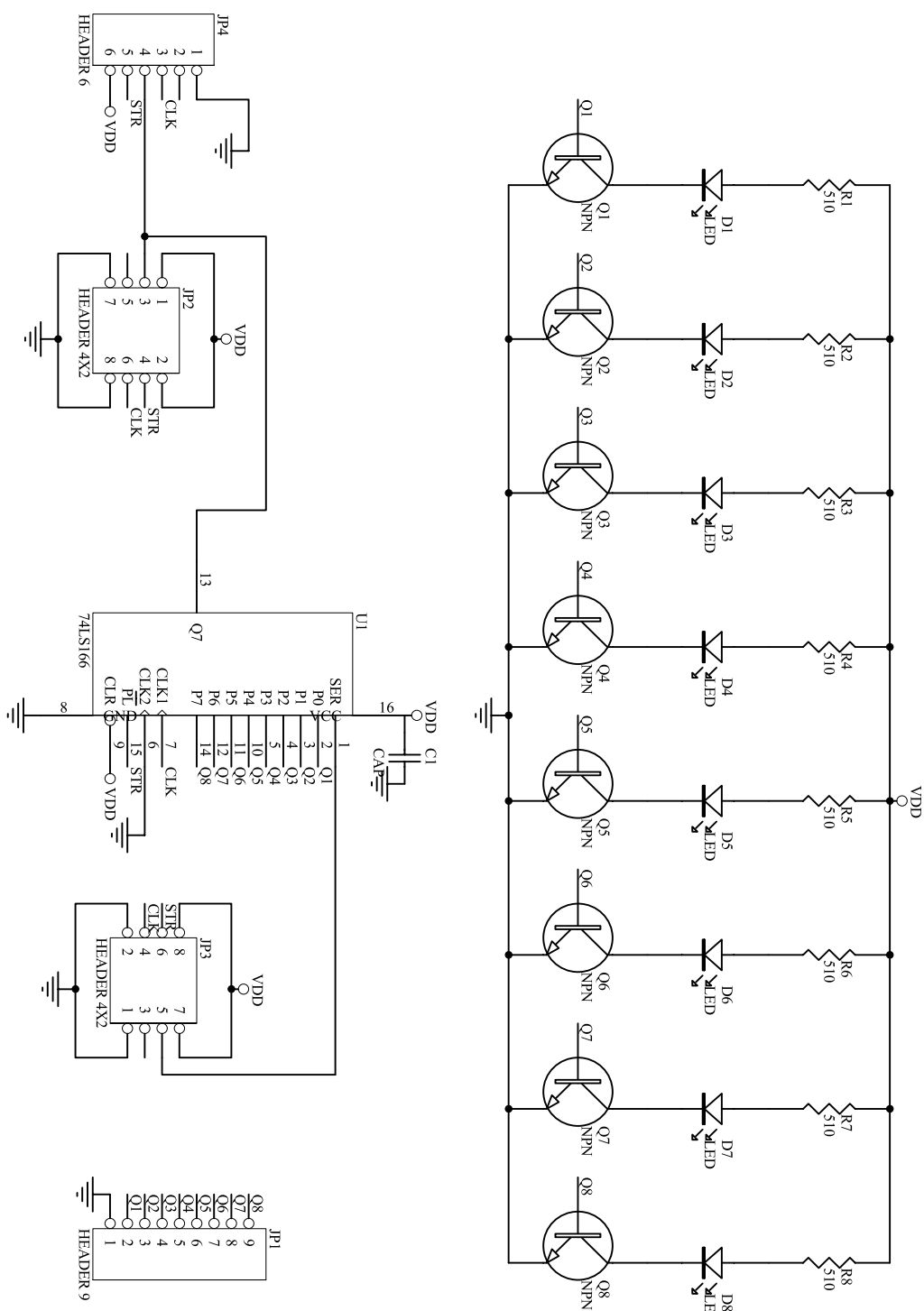


Figure 7. Input Expander
PCB Copper Pattern
(Bottom Layer)

Figure 8. Schematic Diagram of Input Expander Module



SAMPLE CIRCUIT DIAGRAM

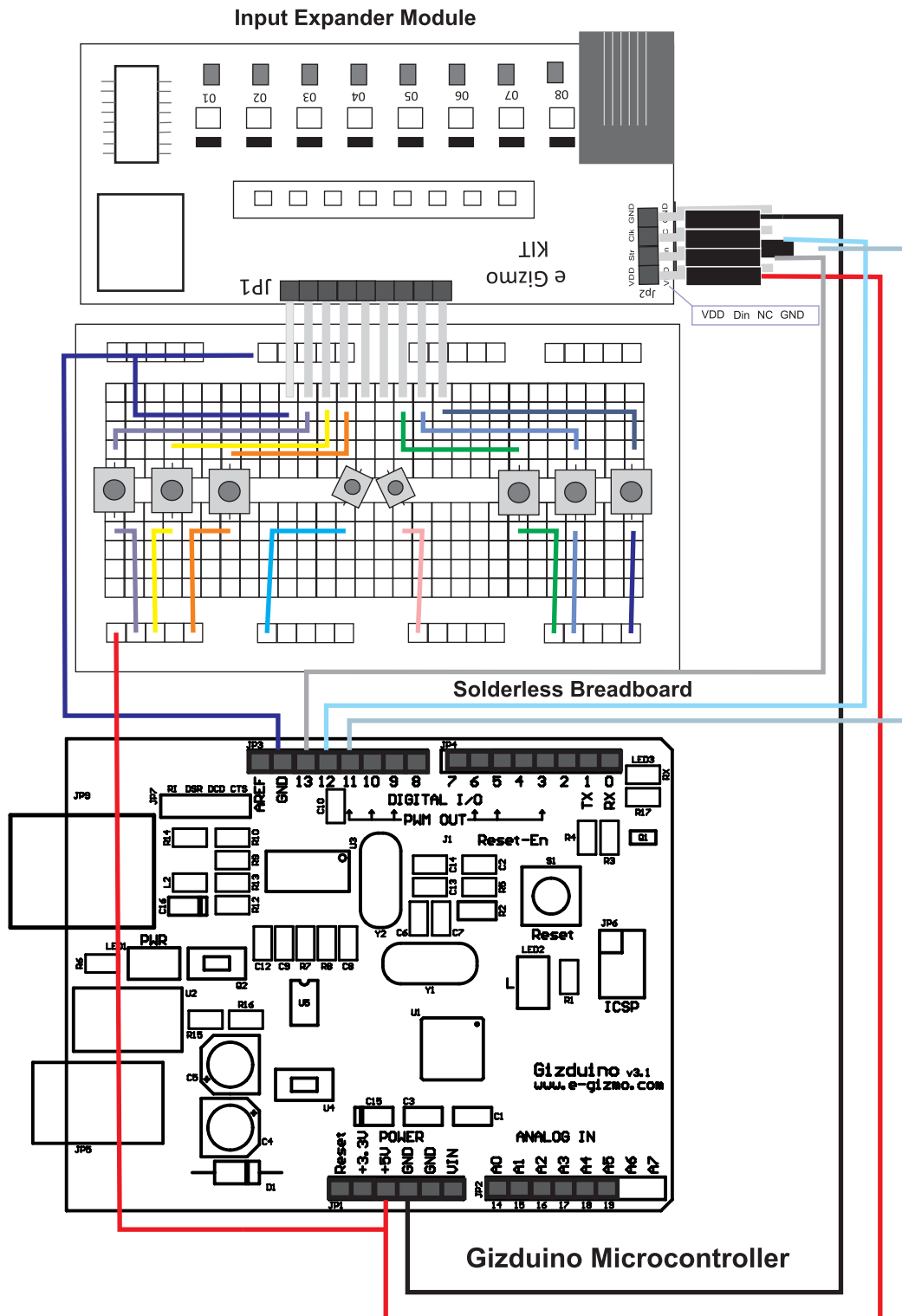


Figure 9.
Sample
Circuit
Diagram

SAMPLE CODES

/ Input Expander Module*

This sample program makes the module a binary counter.

A sample circuit is shown in this folder of how the input expander may be tested independently without

an output expander module.

This test program checks the functionality of the input expander module using the software's serial monitor.

JP2 pin assignments:

Din on digital pin 11

Str on digital pin 12

Clk on digital pin 13

Codes by:

eGizmo Mechatronix Central

Manila, Philippines

April 4, 2013

**/*

// constants won't change. They're used here to

// set pin numbers:

const int INCLK = 13; // input expander CLK pin

const int DIN = 11; // input expander DOUT pin

const int INSTR = 12; // input expander STR pin

const int OUTCLK = A2; // output expander CLK pin

const int DOUT = A1; // output expander DIN pin

const int OUTSTR = A0; // output expander STR pin

// Set delay, bytes, counter, and mask.

int d = 100;

byte byte1;

byte counter;

byte mask;

void setup()

{

// Initialize serial monitor with 9600 baud.

Serial.begin(9600);

// Set pins to output i stands for increment.

pinMode(INCLK, OUTPUT); // Set Clock

in as output

pinMode(DIN, INPUT); // Set Data in as

input

pinMode(INSTR, OUTPUT); // Set Strobe

in as output

pinMode(OUTCLK, OUTPUT); // Set

Clock out as output

pinMode(DOUT, OUTPUT); // Set Data

out as output

pinMode(OUTSTR, OUTPUT); // Set

Strobe out as output

}

// Set clock in cycle

void CLKIN(){

digitalWrite(INCLK, HIGH);

digitalWrite(INCLK, LOW);

}

// Set clock out cycle

void CLKOUT(){

digitalWrite(OUTCLK, HIGH);

digitalWrite(OUTCLK, LOW);

}

// Set strobe in cycle

void STRIN(){

digitalWrite(INSTR, HIGH);

digitalWrite(INSTR, LOW);

}

```
// Set strobe out cycle
void STROUT(){
    digitalWrite(INSTR, HIGH);
    digitalWrite(INSTR, LOW);
}

/// Set LED outputs
void out_LEDs()
{
    for(mask=1; mask>0;)
    {
        if(byte1 & mask)
            digitalWrite(DOUT, HIGH);
        else
            digitalWrite(DOUT, LOW);
        CLKOUT();
        mask = mask << 1;
    }
    STROUT();
}

byte read_LEDs()
{
    byte byte1 = 0;

    digitalWrite(INSTR, LOW);
    CLKIN();
    digitalWrite(INSTR, HIGH);
    for(mask=1; mask>0;)
    {
        if(digitalRead(DIN) == HIGH)
            byte1 |= mask;
        CLKIN();
        mask = mask << 1;
    }
    return byte1; // Returns the value
}

void loop()
{
    byte reading = read_LEDs();
    // Set reading as the variable for LED output
    Serial.println(reading);
}
```