

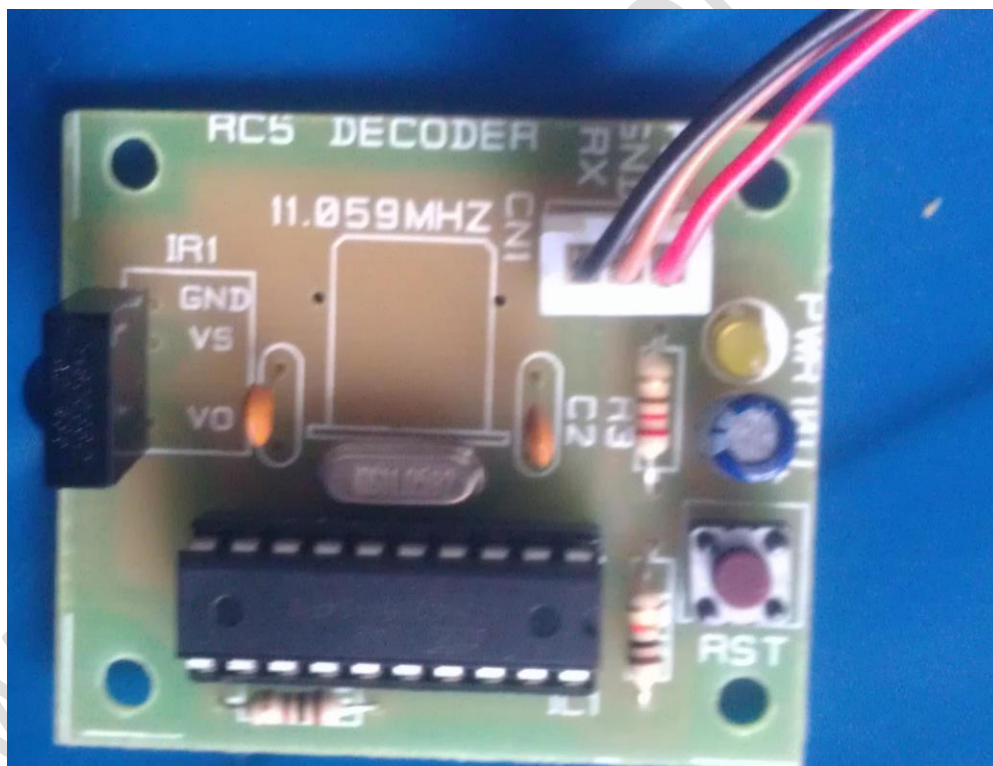
RC5 IR Remote serial Decoder

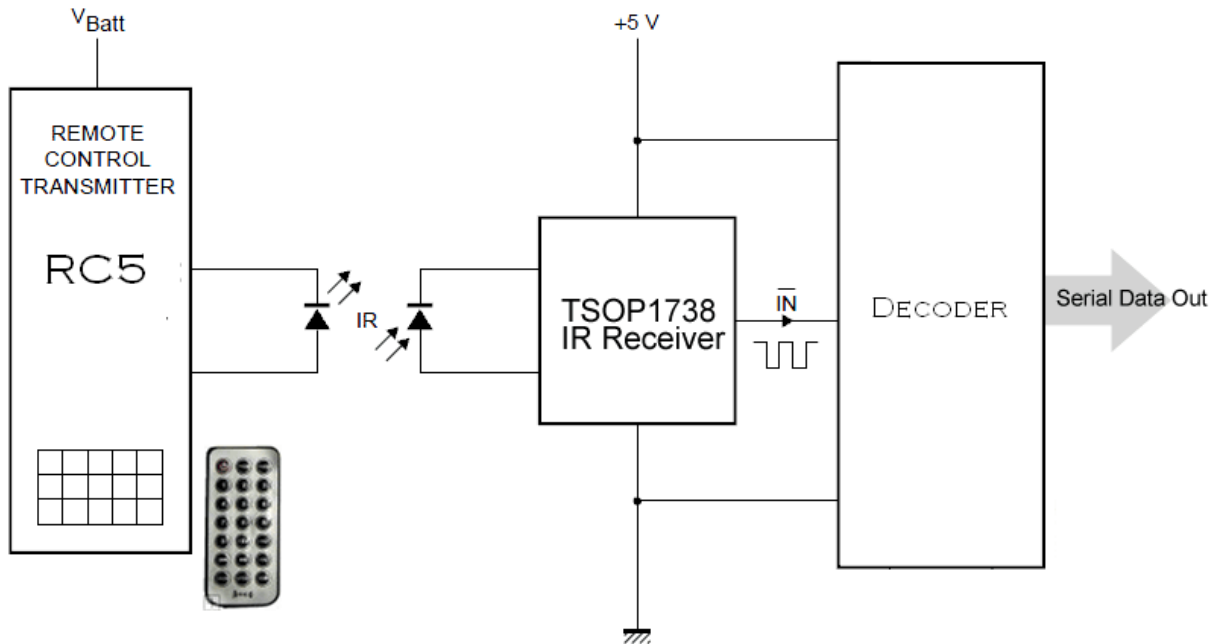
MDU29

(Without remote)

Description

The RC5 remote decoder board is based on the IC AT89c2051 which decodes the received remote control data and output 8 bit serial data output. The decoded data contains various data related to key pressed. This decoded information from transmitter can be used in various ways to make any remote control based applications.





Specifications:

- Simple serial data output
- Compatible with any Microcontroller pins directly
- Very less current consumption 10mA
- Decodes any RC5 (Philips) Remote type
- Baud rate 9600
- Three wire Interface (+5V, Gnd, Tx)

8051 Microcontroller sample code to receive data via Rx pin:

```

// Compiler: Keil

#include <REGX51.H> // standard 8051 defines

// -----

// ----- Hardware Defines -----

// -----

sbit RELAY = P3^7;

char sbuffer[10], ch, toggle, last_toggle;

unsigned char pos;

unsigned int rc5data;

//receive serial character from serial port

char mygetchar(void)
{
    char c;
    while(!RI);
    RI =0;
    c = SBUF;
    return SBUF;
}

// -----

// ----- Main Program -----

// -----

```

```

void main()
{
    // ---- Intialize variables ----
    RELAY = 0; // OFF

    pos = 0;

    // ---- Intialise Serial Port ----

    //Sets up MCU to use 9600 bps @ 11.059 MHz Crystal

    SCON = 0x52; // 8-bit UART mode

    TMOD = 0x20; // timer 1 mode 2 auto reload

    TH1= 0xfd; // 9600 8-n-1

    TR1 = 1; // run timer1

    last_toggle = -1;

    // ---- Program Loop ----
    while(1)
    {
        ch = mygetchar(); //loop till character received

        if(ch==0x0A) // if received character is <LF> end of line, time to display
    {
        pos = 0; // buffer position reset for next reading

        // extract rc5 data from serial buffer to 16 bit value

        rc5data = ((sbuffer[2]-'0')<<4)|(sbuffer[3]-'0'); // convert data from ASCII to low byte

```

rc5data |= (((sbuffer[0]-'0')<<4)|(sbuffer[1]-'0')<<8); // convert data from ASCII to high
byte

```
        if((rc5data&0x0800)==0x0800) // check toggle bit from rc5 data is set
    {
        toggle = 1;
    } else
    {
        toggle = 0;
    }

    if(toggle!=last_toggle) // check if new key is pressed
    {
        last_toggle = toggle; // store value for next time compare

        switch(rc5data&0x003F) // mask other bits and compare command bytes
        {
            case 0x01: // Key1 value
                if(RELAY==0) // toggle relay
                    RELAY = 1;
            else
                RELAY = 0;

            break;
        }
    }
}
```

```
case 0x02:
    // Key2 pressed
    break;
case 0x03:
    // Key3 pressed
    break;
}
}
} else { //store serial data to buffer
    sbuffer[pos] = ch;
    pos++;
}
} // end while
} // end main
```