

```
int modbus_addr = 0x06; // PUT THE FINAL HEX VALUE THAT WILL BE TRANSMITTED
int read_coil = 0x01; // READ COIL
int no_bytes = 0x01; // NO. OF BYTES
//int data0_hi = 0x10; // DATA TO BE TRANSMITTED
int data0_lo = 0x00; // DATA TO BE TRANSMITTED
```

```
int pushButton2 = 2;
int pushButton3 = 3;
int pushButton4 = 4;
int pushButton5 = 5;
int pushButton6 = 6;
int pushButton7 = 7;
```

```
byte button2 = 0x00;
byte button3 = 0x00;
byte button4 = 0x00;
byte button5 = 0x00;
byte button6 = 0x00;
byte button7 = 0x00;
```

```
void setup() {
```

```
    Serial.begin(9600);
    pinMode(pushButton2, INPUT);
    pinMode(pushButton3, INPUT);
    pinMode(pushButton4, INPUT);
    pinMode(pushButton5, INPUT);
    pinMode(pushButton6, INPUT);
    pinMode(pushButton7, INPUT);
```

```
}
```

```
void loop() {
```

```
int buttonState2 = digitalRead(pushButton2);
int buttonState3 = digitalRead(pushButton3);
int buttonState4 = digitalRead(pushButton4);
int buttonState5 = digitalRead(pushButton5);
int buttonState6 = digitalRead(pushButton6);
int buttonState7 = digitalRead(pushButton7);
```

```
//-----
```

```
//---- PUSH BUTTON 2 LOGIC CONVERSION
```

```
if(buttonState2 == 1){
button2 = 0x01;
}
else{
button2 = 0x00 ;
}
}
```

```

//-----
//---- PUSH BUTTON 3 LOGIC CONVERSION

if(buttonState3 == 1){
button3 = 0x02;
}
else{
button3 = 0x00 ;
}
//-----
//---- PUSH BUTTON 4 LOGIC CONVERSION
if(buttonState4 == 1){
button4 = 0x04;
}
else{
button4 = 0x00 ;
}
//-----
//---- PUSH BUTTON 5 LOGIC CONVERSION

if(buttonState5 == 1){
button5 = 0x08;
}
else{
button5 = 0x00 ;
}

//-----
//---- PUSH BUTTON 6 LOGIC CONVERSION
if(buttonState4 == 1){
button6 = 0x10;
}
else{
button6 = 0x00 ;
}
//-----
//---- PUSH BUTTON 7 LOGIC CONVERSION

if(buttonState7 == 1){
button7 = 0x20;
}
else{
button7 = 0x00 ;
}
//--FINAL BYTE COMPILATION -----

//-----
byte final_byte = byte(button2 | button3 | button4 | button5 | button6 | button7);
data0_lo = final_byte;
//-----

```

```

// LRC ALGO
// LRC ALGORITHM BEGINS HERE
int add_lrc = ( modbus_addr + read_coil);
int add_lrc_1 =0X0FF & add_lrc ;
int add_lrc_2 = (add_lrc_1 + no_bytes);
int add_lrc_3 = 0X0FF & add_lrc_2 ;
int add_lrc_4 = (add_lrc_3 + data0_lo);
int add_lrc_5 = 0XFF & add_lrc_4 ;

int ff_lrc = (0xFF - add_lrc_5);

int lrc_final_a = ((ff_lrc) + 1) ;
String lrc_final = String(lrc_final_a,HEX);
lrc_final.toUpperCase();
//-- LRC CALCULATION ENDS HERE
//-----
//-----
//modbus address msb
int modbus_addr_hi_a = (0xF0 & modbus_addr);
int modbus_addr_hi_b = ((modbus_addr_hi_a)/16);
String modbus_addr_hi_c = String((modbus_addr_hi_b),HEX);
modbus_addr_hi_c.toUpperCase();
//modbus address lsb
int modbus_addr_lo_a = (0X0F & modbus_addr);
String modbus_addr_lo_c = String((modbus_addr_lo_a),HEX);
modbus_addr_lo_c.toUpperCase();
//-----
//modbus read coil msb
int read_coil_hi_a = (0xF0 & read_coil);
int read_coil_hi_b = ((read_coil_hi_a)/16);
String read_coil_hi_c = String((read_coil_hi_b),HEX);
read_coil_hi_c.toUpperCase();
//modbus read coil lsb
int read_coil_lo_a = (0X0F & read_coil);
String read_coil_lo_c = String((read_coil_lo_a),HEX);
read_coil_lo_c.toUpperCase();
//-----
//number of bytes msb
int no_bytes_hi_a = (0xF0 & no_bytes);
int no_bytes_hi_b = ((no_bytes_hi_a)/16);
String no_bytes_hi_c = String((no_bytes_hi_b),HEX);
no_bytes_hi_c.toUpperCase();
//number of bytes lsb
int no_bytes_lo_a = (0X0F & no_bytes);
String no_bytes_lo_c = String((no_bytes_lo_a),HEX);
no_bytes_lo_c.toUpperCase();
//-----
//DATA 0 HI MSB
//int data0_hi_hi_a = (0xF0 & data0_hi);

```

```

//int data0_hi_hi_b = ((data0_hi_hi_a)/16);
//String data0_hi_hi_c = String((data0_hi_hi_b),HEX);
//data0_hi_hi_c.toUpperCase();
////DATA 0 HI LSB
//int data0_hi_lo_a = (0X0F & data0_hi);
//String data0_hi_lo_c = String((data0_hi_lo_a),HEX);
//data0_hi_lo_c.toUpperCase();
//-----
//DATA 0 LO MSB
int data0_lo_hi_a = (0XF0 & data0_lo);
int data0_lo_hi_b = ((data0_lo_hi_a)/16);
String data0_lo_hi_c = String((data0_lo_hi_b),HEX);
data0_lo_hi_c.toUpperCase();
//DATA 0 LO LSB
int data0_lo_lo_a = (0X0F & data0_lo);
String data0_lo_lo_c = String((data0_lo_lo_a),HEX);
data0_lo_lo_c.toUpperCase();
//-----

// COOMAND FROM SERVER ASCII RAW : 3A 30 36 30 31 30 30 36 41 30 30 30 44 38 32
0D 0A
String modbus_final_txmt =( ":" + String(modbus_addr_hi_c) +
String(modbus_addr_lo_c) + String(read_coil_hi_c) + String(read_coil_lo_c) +
String(no_bytes_hi_c) + String(no_bytes_lo_c) + String(data0_lo_hi_c) +
String(data0_lo_lo_c) + String(lrc_final) ) ;
Serial.println(modbus_final_txmt); // crlf
//Serial.println(lrc_final); ONLY FOR LRC TEST
// Serial.println(":0601021000E7"); // crlf
Serial.println(button3);
Serial.println(buttonState3);
Serial.println(final_byte);
//LRC CALCULATE:
//ADD ALL HEX BYTES = 9F 1001 1111
//1111 1111 - 1001 1111 = 0110 0000 NOW ADD 1 THEN 0110 0001 = 61
// DONES

delay(300); // delay in between reads for stability
}

```