

```

// _250226_C_Solax_RequestData
// Request data from Solax X1 and extract powervalue from the data packet
// Try-out with registered address

// Arduino MAX 485                SOLAX
// 5  DE Data Enable (DEPin)
// 4  RE Receive Enable(REPin)
// 3  RO Receive Out (SSerialRX)
// 6  DI Data In (SSerialTX)
// 3.3V VCC (5V also allowed)
// GND GND
// RS485 A RJ45 4
// RS485 B RJ45 5

#include <SoftwareSerial.h>
#define SSerialRX 3 // Receive Out
#define REPin 4 // Receive Enable
#define DEPin 5 // Data Enable
#define SSerialTX 6 // Data In
#define RS485Receive LOW
#define RS485Transmit HIGH

SoftwareSerial RS485Serial(SSerialRX, SSerialTX);
int SolaxPower = 0;
byte InByte = 0;
byte byteInput[63];
byte RequestData[]={0xAA,0x55, 0x00,0x00, 0x00,0x0A , 0x11, 0x02, 0x00, 0x01,0x1C};
// Header , Source , SOLAX ,Control,Func,Length, Check 2b

void setup()
{
  Serial.begin(9600);
  Serial.println("_250226_C_Solax_RequestData");
  Serial.println();
  pinMode(REPin, OUTPUT);
  pinMode(DEPin, OUTPUT);
  pinMode(SSerialRX, INPUT);
  pinMode(SSerialTX, OUTPUT);
  digitalWrite(REPin, RS485Receive);
  digitalWrite(DEPin, RS485Receive);
  RS485Serial.begin(9600); // set data rate
}

void GetSolaxData()
{ // reads the datasstream from the Solax and updates rhe delivered power

  int index = 0;
  while(RS485Serial.available())
  {

```

```

InByte = (byte)RS485Serial.read();
byteInput[index] = InByte;
index++;
}
for(int i=0;i<63;i++)
{
Serial.println();
Serial.print(i);
Serial.print(" Dec: ");
Serial.print(byteInput[i]);
Serial.print(" HEX: ");
Serial.print(byteInput[i],HEX);
}
SolaxPower = byteInput[28] + 256 * byteInput[27];
}

void loop()
{
// * * * SEND A DATA REQUEST TO THE INVERTER * * *
digitalWrite(REPin, RS485Transmit);
digitalWrite(DEPin, RS485Transmit);
RS485Serial.write(RequestData,sizeof(RequestData));

// * * * SWITCH TO RECEIVE MODE AND UPDATE THE ACTUAL DELIVERED POWER * * *
digitalWrite(REPin, RS485Receive);
digitalWrite(DEPin, RS485Receive);
if(RS485Serial.available())
{
GetSolaxData();
Serial.println();
Serial.print("Watts: ");
Serial.print(SolaxPower);
}

// * * * DO THE SWITCHING OF THE SOCKETS ACCORDING THE UPDATED POWER * * *

delay(10000); // 10 sec as trial setting
}
/*
Output
_250226_C_Solax_RequestData

0 Dec: 170 HEX: AA
1 Dec: 85 HEX: 55
2 Dec: 0 HEX: 0
3 Dec: 10 HEX: A
4 Dec: 0 HEX: 0
5 Dec: 0 HEX: 0
6 Dec: 17 HEX: 11

```

7 Dec: 130 HEX: 82  
8 Dec: 52 HEX: 34  
9 Dec: 0 HEX: 0  
10 Dec: 36 HEX: 24  
11 Dec: 0 HEX: 0  
12 Dec: 13 HEX: D  
13 Dec: 10 HEX: A  
14 Dec: 120 HEX: 78  
15 Dec: 0 HEX: 0  
16 Dec: 0 HEX: 0  
17 Dec: 0 HEX: 0  
18 Dec: 30 HEX: 1E  
19 Dec: 0 HEX: 0  
20 Dec: 0 HEX: 0  
21 Dec: 0 HEX: 0  
22 Dec: 32 HEX: 20  
23 Dec: 9 HEX: 9  
24 Dec: 122 HEX: 7A  
25 Dec: 19 HEX: 13  
26 Dec: 134 HEX: 86  
27 Dec: 2 HEX: 2  
28 Dec: 255 HEX: FF  
29 Dec: 255 HEX: FF  
30 Dec: 255 HEX: FF  
31 Dec: 0 HEX: 0  
32 Dec: 1 HEX: 1  
33 Dec: 66 HEX: 42  
34 Dec: 145 HEX: 91  
35 Dec: 0 HEX: 0  
36 Dec: 0 HEX: 0  
37 Dec: 53 HEX: 35  
38 Dec: 233 HEX: E9  
39 Dec: 0 HEX: 0  
40 Dec: 2 HEX: 2  
41 Dec: 0 HEX: 0  
42 Dec: 0 HEX: 0  
43 Dec: 0 HEX: 0  
44 Dec: 0 HEX: 0  
45 Dec: 0 HEX: 0  
46 Dec: 0 HEX: 0  
47 Dec: 0 HEX: 0  
48 Dec: 0 HEX: 0  
49 Dec: 0 HEX: 0  
50 Dec: 0 HEX: 0  
51 Dec: 0 HEX: 0  
52 Dec: 0 HEX: 0  
53 Dec: 0 HEX: 0  
54 Dec: 0 HEX: 0  
55 Dec: 0 HEX: 0

56 Dec: 0 HEX: 0  
57 Dec: 0 HEX: 0  
58 Dec: 0 HEX: 0  
59 Dec: 15 HEX: F  
60 Dec: 4 HEX: 4  
61 Dec: 8 HEX: 8  
62 Dec: 227 HEX: E3  
Watts: 767  
\*/  
\*/