



HOW TO MANUFACTURE

EnAccess Open-Source Meter Project

[Abstract](#)

This document summarizes the manufacturing guide to enable a skilled person to manufacture the smart energy meter.

First Electric

info@firstelectricco.com



Table of Contents

1. WEB SOFTWARE	Error! Bookmark not defined.
2. HARDWARE	3
3. CALIBRATION	Error! Bookmark not defined.
4. FIRST TRIAL	6



Web Software :

The web software is split into different parts as highlighted below.

- i. Extraction of zip file from Github.
- ii. Creating database.
- iii. Linking Thingsboard public link to webpage.

Extraction of zip file from Github :

The web software can be replicated by developer downloading the zip code file from <https://github.com/EnAccess/OpenSmartMeter/tree/main/Web%20software>. after the code is downloaded, developer will need to host the web on a hosting platform.

Creating database :

The database schema file can be downloaded from the Github link provided below

<https://github.com/EnAccess/OpenSmartMeter/tree/main/Web%20software/Database%20schema> for testing purpose,, next user can assign a meter number as desired from MT1 to MT99990 and insert into the meter Database column for meter number.

Developer need to change password and database name in web code to same password and database name as created by developer to enable access to the database.

Linking Thingsboard public link to webpage :

After Thingsboard page is setup, developer make the data available by changing the privacy



setting from private to public, next the public link is copied to the Thingsboard column specified in the database for each user.

Hardware :

The hardware is splitted into different parts as highlighted below.

- a. Extracting manufacturing file.
- b. Soldering/assembling PCB.

Extracting manufacturing file :

The first step is to extract the manufacturing/production file from Github in the link provided below

<https://github.com/EnAccess/OpenSmartMeter/tree/main/Meter%20hardware%20design/production%20file>, the zip file is sent to a PCB manufacturing company for the board to be manufactured. The manufactured PCB is shipped back to the design engineer.

Soldering/assembling PCB :

The PCB is assembled, soldered according to the components value as labeled on the silkscreen layer of the PCB board. The turn ratio of the ferrite core transformer is given below in figure 1.

Feed back turn = 7turns.

Input (100v - 270v) = 53 turns.

Output(12v) = 10 turns.

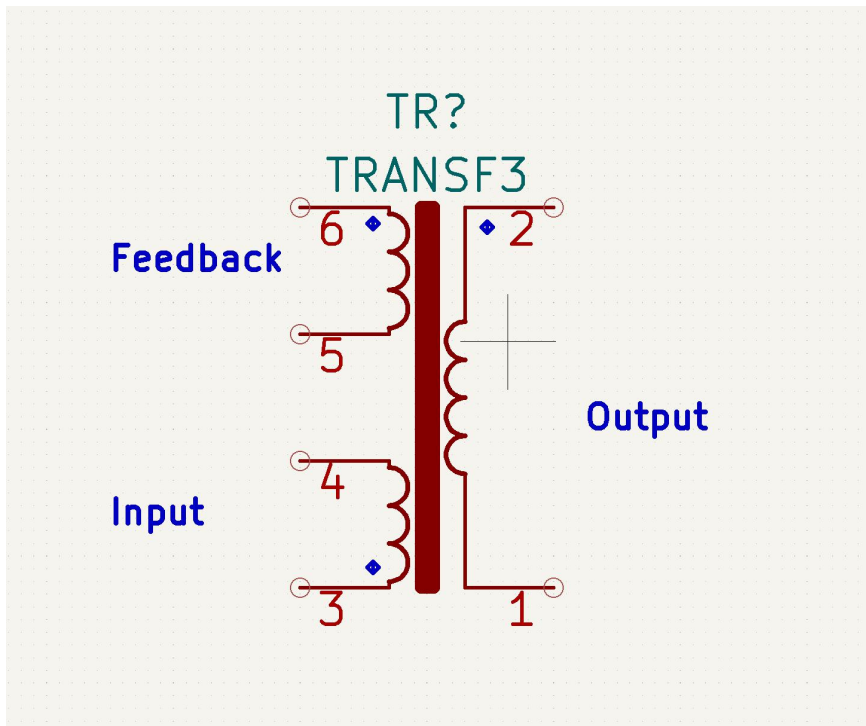


Figure 1 : Transformer configuration/turn diagram.

Calibration :

The meter calibration is done after the meter PCB component is fully assembled alongside exterior casing.

Step 1 : Calibrating voltage and current

The design engineer proceed to calibrate the voltage, current. The voltage and current is calibrated by calculating the value measured by meter as compared to the true value measured by a test bench, the excel sheet available in the link provided below is used to calibrate the voltage and current by inputting the measured and actual values of voltage and current in the appropriate cell of the excel calculator.

<https://github.com/EnAccess/OpenSmartMeter/blob/main/Documentations/Datasheet/Energy%20setpoint%20calculator.xlsx>, after this is done, the hex code gotten is written to the



necessary registers of **Ugain**, **IgainN** in the SAM_UART.cpp library provided in the link below.

https://github.com/EnAccess/OpenSmartMeter/blob/main/Firmware%20code/Library/SamATM90E26_library.zip after this is done, the checksum2 value displayed on the LCD screen while the meter is starting up is written to the **CSTwo** register in the SAM_UART.cpp library, after this is done, design engineer re-uploads the code to the chip and all measuring parameter is correct.

Note : a meter test bench is needed to know the true value of voltage and current to be calibrated into the meter.

Step 2 : setting impulse rate

The impulse rate is changed by writing to PLconstH and PLconstL in the SAM_UART.cpp library after inputting the correct value of

$U_n = 240V$ (measured voltage).

$I_b = 5A$ (base current).

$G_L = 1$ (line current circuit gain).

$V_L = 21.3$ (sampling voltage of current circuit in MV).

$V_U = 260$ (sampling voltage of voltage circuit in MV).

$M_C = 2000$ (impulse rate) **Note:** the impulse is 1000, however the value is multiplied by 2 due to current scaling.

Note : the values specified above is the values used according to the resistor value used in meter PCB.



Step 3 : calibrating impulse

A test bench is required to measure the % error of the meter, after this is done the design engineer can input the error gotten into the excel calculator provided in the link below.

<https://github.com/EnAccess/OpenSmartMeter/blob/main/Documentations/Datasheet/Energy%20setpoint%20calculator.xlsx>, after the error is calculated, the design engineer write the value gotten for Lgain and Igain to the necessary register in the SAM_UART.cpp library, then the new value of checksum1 is rewritten to the library for the meter to blink properly.

First trial :

STEP 1 : testing component proper assembling and soldering

After developer is done assembling the components on the PCB, it is advisable to test the PCB first by connecting the battery to the battery connector to test if all connection is correct or not, if all connection is correct the meter LCD will display, then meter boot and try connecting to internet before the LCD is turned off due to unavailability of mains and to preserve battery life.

STEP 2 :

Connect the input of the meter through an incandescent bulb without any load to the meter to prevent the components from going up in smoke if there is a short or mis-soldering, if the meter PCB is faulty the incandescence bulb will light up brightly, if all is ok, the meter will start up however the bulb may light up a little bit if all is ok.

Warning : pls do not assemble or try manufacturing the meter if you are not a skilled electronics engineer or engineer with similar skill.