

# Low cost power meter customization manual

17th March 2017

## Contents

<b>1</b>	<b>Requirements</b>	<b>1</b>
<b>2</b>	<b>Software an drivers</b>	<b>1</b>
2.1	FT232R USB UART . . . . .	1
2.2	USB2.0-Serial . . . . .	2
2.3	POWduino Software . . . . .	2
<b>3</b>	<b>Customization</b>	<b>5</b>

## 1 Requirements

- Home build arduino based power meter prototype ("POWduino").
- Photodiode
- Computer

At the moment one prototype is available (ask Sven). The photodiode needs to be connected either by BNC or SMA to the prototype. *Hint:* Bernhard built some very nice large area photodiodes with SMA connector.

## 2 Software an drivers

You don't need the arduino software on your computer. But depending on the origin of your arduino you might need some drivers. I discovered two cases.

### 2.1 FT232R USB UART

If there is an unknown hardware called "FT232R USB UART" in the device manager like shown in Figure 1, you need to install a virtual COM port driver. You can find it on the AFS server<sup>1</sup>.

<sup>1</sup>[/afs/iqo.uni-hannover.de/group/morgner/Software/Laborgeraete/POWduino/driver/CDM21226\\_Setup.exe](http://afs/iqo.uni-hannover.de/group/morgner/Software/Laborgeraete/POWduino/driver/CDM21226_Setup.exe)

## 2.2 USB2.0-Serial

If there is an unknown device called "USB2.0-Serial" you need to install a different driver (Figure 2). This was the case for me, when I was using an arduino compatible board instead of an original arduino. You can also find the driver on the AFS server<sup>2</sup>.

## 2.3 POWduino Software

When the arduino is recognized by your computer you should see an additional COM-port in the device manger. Afterwards, you need to install the POWduino Software. You can find it on the AFS server<sup>3</sup>. When you start the program you will see the window shown in Figure 3. Select the COM port to which the arduino is connected in the "Serialport" choice box. A configuration file is not necessary. It gives you the opportunity to load a saved calibration file. After pressing the "OK" button the main window will appear. It will look like the one shown in Figure 4. Without calibration you can continue to figure out, which parts of the electronic circuit you need to change to work with your system. In this case continue with section 3.

If you already have a customized version of the POWduino, which is working with your setup and has the dynamic range optimized you can calibrate the values measured by the arduino to an arbitrary power scale. To do so, click *File-Calibrate*. You will see a window like shown in Figure ???. In the upper half you see a table in which you have to insert the measured value of the arduino and the corresponding power (measured with a well working power meter). To improve the accuracy of the  $P(U)$  fit you can add as many points as you wish by pressing the "+" button. The "-" button reduces the points for the fit. In the lower part of the window you can choose the degree of the polynomial to fit to your measured values. It should be mentioned, that you need at least  $n + 1$  points to fit a polynomial of degree  $n$ . After you are done press the "Confirm" button and you will see that the values on the screen now has the unit watt. If you want to save your calibration click *File-Save*. To load a saved calibration file click *File-Open*. Alternatively you can load the calibration file already in the initialization window.

---

<sup>2</sup>/afs/iqo.uni-hannover.de/group/morgner/Software/Laborgeraete/POWduino/driver/CH341SER/

<sup>3</sup>/afs/iqo.uni-hannover.de/group/morgner/Software/Laborgeraete/POWduino/software/

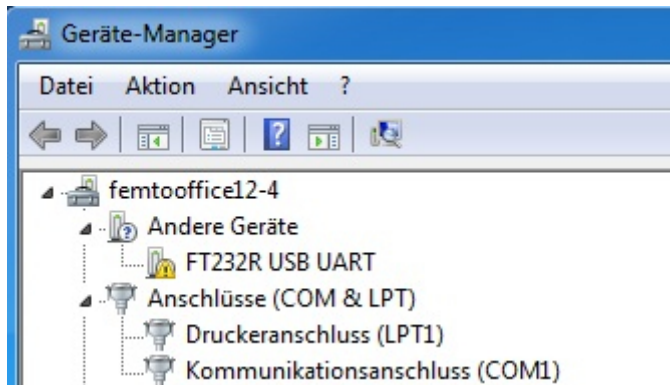


Figure 1: Screenshot of windows device manager. In this case the virtual COM port driver is missing.

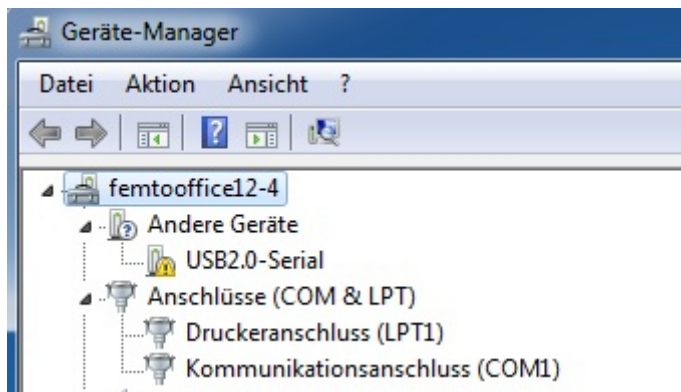


Figure 2: Screenshot of windows device manager. In this case the CH341SER driver is missing.

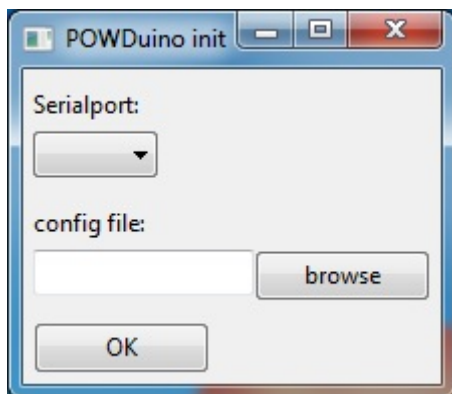


Figure 3: Screenshot of the initialization window of the software.

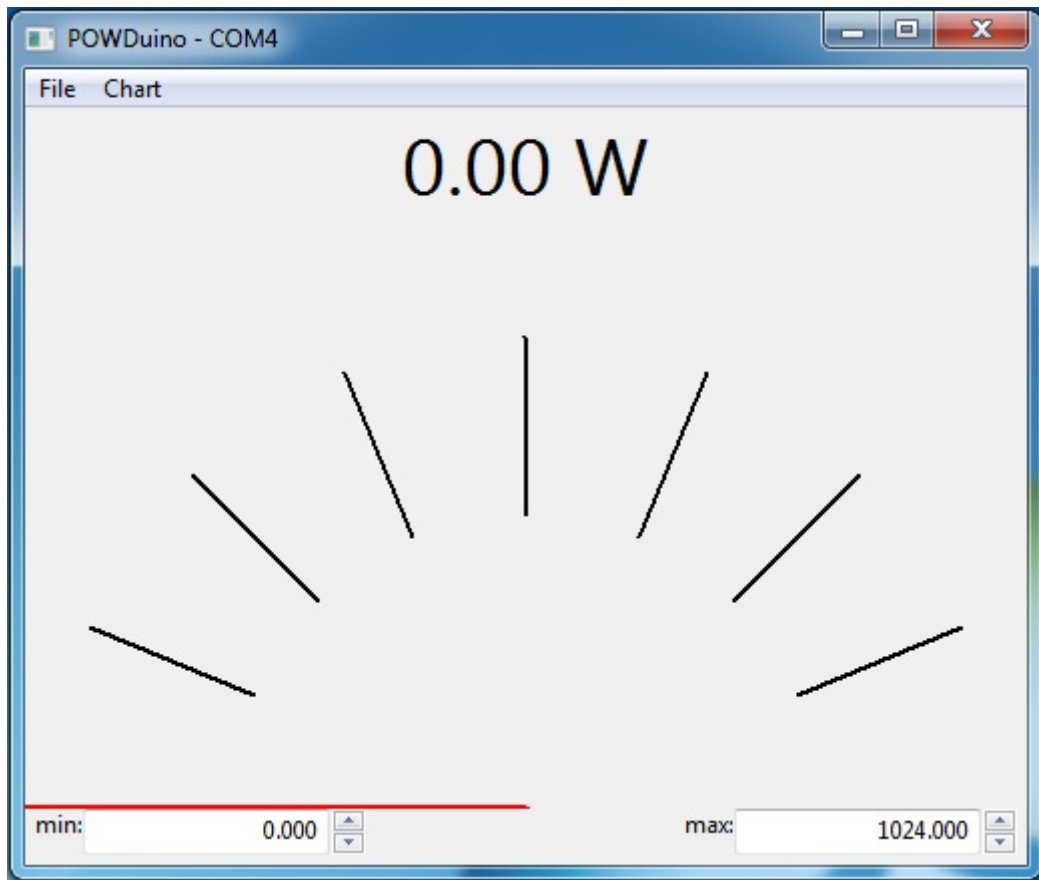


Figure 4: Screenshot of the main window of the POWduino software.

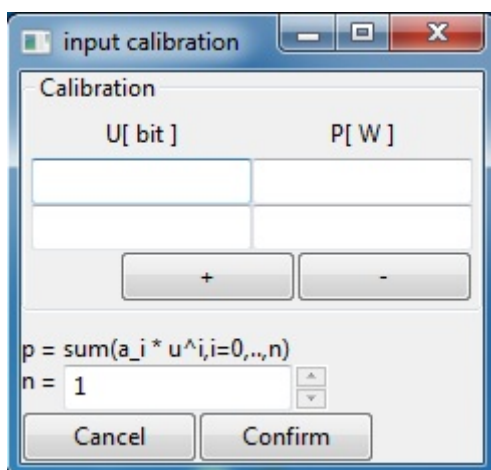


Figure 5: Screenshot of the calibration window of the POWduino software.

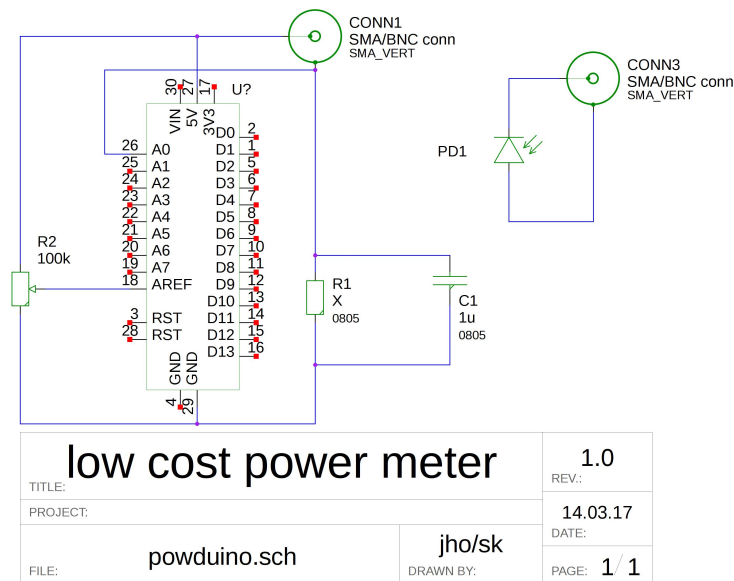


Figure 6: Circuit diagram of the POWduino.

### 3 Customization

The prototype has a resistor R1, which can be replaced (see Figure 6 and 7). This resistor has to be matched to the photodiode and the expected signal intensity. To do so, connect the photodiode to the POWduino and run the software. If you illuminate the photodiode the value shown from the program could be between 0 and 1024. If the power meter is saturating (showing 1024) try to change the state of the potentiometer R2 (see Figure 6 and 7). With this one you can fine tune the dynamic range of the power meter. If you are not able to avoid saturation with the potentiometer replace R1 with a smaller resistor (R1= 470 Ω was working for me to measure the leakage of 1.5 W behind a mirror).

After you have found a resistor R1, which is not leading to saturation, you can fine tune the dynamic range with R2. Note the value of R1, which works for you and build your own power meter, following the plan shown in Figure 6.

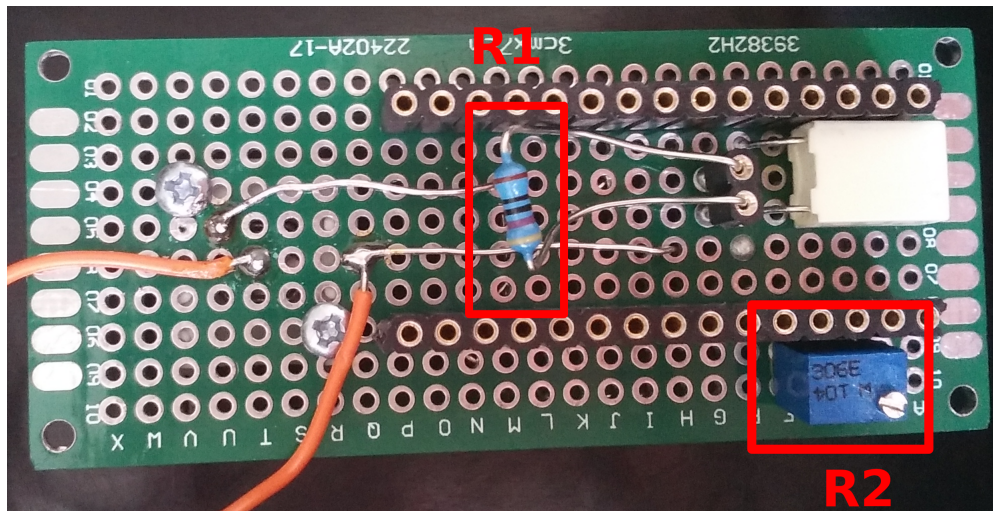


Figure 7: Picture of the POWduino prototype.