Solar charge controller (with MPPT algorithm and BT communication) | Article overview

**Article outline:**

1. Introduction to the story. Shortly about my background;
2. Device design and production flow/timeline, my approach to the project;
3. Requirements, assumptions, pre-design research etc.
4. Hardware – from prototypes to custom made PCBs
   1. Introduction, hardware requirements etc.
   2. Switching step-down regulator
   3. MCU & Bluetooth module
   4. PCB design and soldering
5. Software – from first sketches to C++ board library
   1. Software requirements
   2. Final approach
6. Callibration and testing
7. Conclusion and future development plans
8. Refrences

**Article requirements/assumptions:**

1. No. of characters: 10 000 – 25 000
2. No. of photos: ~ 10 – 20
3. At least one (ideally two) videos, one presenting final effect

DIY Solar charger with bluetooth communication

Each year renewable energy sources, with photovoltaic panels at the forefront, increase their contribution to the overall electricity production. The view of the solar installation on the roof of our neighbors does not surprise anyone anymore, but the way such a system works is not so widely known. The Solar charger project described below is, firstly, a technical challenge for a student as myself, but secondly, it’s an attempt to understand the principles of operation of the technology that is shaping the future of energy generation. How to charge a battery from the solar panel? Why MPPT is important especially for bigger instalations? And what actually is MPPT algorithm? …

A little bit of background

In September last year, I’ve started a bachelor degree in electronics at the University of Manchester. I’m from Poland, so I was excited about moving out to the UK and studying the subject, which I have always been fascinated with. Unfortunately quickly it turned out that there was no chance for any classes at the campus, so I was forced to move back to Poland and start studying remotely. I decided however, that regardless of the situation, I will try to make the most of the theoretical content provided by the lecturers and complement this with my own practical projects. And that’s what I’ve done.

Project definition – my approach to handle technical challenges

After the first semester and an introductory course on semiconductors I’ve decided to try my hand out at solar energy and build my own basic system comprising of everything I need to generate, convert, store and use the energy provided by the Sun. As a project general requirement I’ve also decided that I will try to keep it well organised and follow simple process that I have developed based on Agile / Lean management approach (figure 1).

*Figure 1 - Hardware development process*

Figure 1 – Hardware development process

Apart from following a predetermined plan I’ve also decided to use version control system to back-up all files regularly and keep track of any changes to the design. For this I used GitHub with their desktop app. You can find full repository and design files here.

I have started with general system requirements and pre-design research. After digging a bit in the online library resouces I found a couple of great resources:

* "Photovoltaic Power System: Modeling, Design and Control" by Weidong Xiao (University of Sydney) [this was my main source of information about Photovoltaic technology]
* “Solar MPPT Battery Charger for the Rural Electrification System” by Microchip (36 p. application note) [I have used it a bit as an example of how such system can be implemented]
* Other datasheets, application notes and online examples of similar projects (e.g. [Arduino MPPT Solar Charge Controller](https://www.instructables.com/ARDUINO-SOLAR-CHARGE-CONTROLLER-Version-30/))

At this point I have also specified general system requirements such as:

* The system will comprise of 10 W monocrystalline solar panel and 7 Ah (12 V) rechargeable sealed lead acid battery (mainly for proof-of-concept purposes);
* The device should be able to act as a step-down converter with adjustable output voltage in the range of 13.0 – 15.0 V;
* There should be an option for sending reports on the state of the device wirelessly over bluetooth;
* The charger needs to be able to track maximum power point (MPPT algorithm);

Hardware – from prototypes to custom made PCBs

The project of the charger can be divided into two main parts: hardware and software (or rahter firmware). The hardware on the other hand comprises of two sections: Switching step-down regulator and MCU & Bluetooth section.

**Adjustable buck (step-down) converter**

It quickly turned out that if I want to charge battery, especially from the Solar Panel the core of my hardware will be an adjustable buck converter. I have used such of-the-shelf modules in the past, however I have never tried to build one myself. I started with the aim of building a converter from scratch, based on MOSFETs and fundamental electronic components but after few attempts I discovered that having only very basic circuit analysis experience it’s close to impossible for me to make it working and moreover efficient. Hence, I decided to use an IC manufactured exactly for that purpose.

I choosen LM2576-ADJ with a output voltage of 1.23 to 37 V (±4%) and guaranteed 3.0 A output current working at the frequency of 52 kHz. It requires couple of additional components such as electrolitic capacitors, resistors, Schottky diode and an inductor however the datasheet guide the user quite well throughout the component selection process.

Apart from LM2576 chip and supporting components I’ve added also reverse current protection diode to prevent current flowing back from the battery to the solar panel, e.g. during the night. There are also two voltage dividers for measurements, current sensor header along with a couple of testpoints and jumpers (may also be used for connecting external switches).

For the buck converter to be really digitally adjustable I created a feedback loop that is connected to MCU (Microcontroller unit), such that by changing FDBK voltage output voltage can be controlled.

*Figure 2 - LM2576 step-down schematics*

**Microcontroller unit and bluetooth module**

While the charger is mainly the buck converter it has to be controlled by some external subsystem. In the case of this project I decided to use Atmega328P 8-bit AVR microcotroller, as I had experience with Arduino boards, that are in a lot of cases based on AVR MCUs. It has internal ADC (used for voltage measurements) as well as hardware and software UART serial ports (for debuging and bluetooth communication).

One of the idea of the device was to include in it some sort of wireless communication so that you can check the state of the charger without connecting a cable to the

For Digital-analog-converter I have used external MCP4725 DAC board with I2C interface. I tried to create internal on-board DAC using PWM signal from the Atmega microcontroller alongside with operational amplifier and RC filter however its performance was not satisfactory enough to controll the feedback loop of the buck converter.

**PCB design and soldering**

After about two months of prototyping (“Design 🡪 Build 🡪 Test” loop iterations) and finding the most suitable components, modules and interconnections between all of them, it came time for finalising design files. I’ve made some last changes to the schematic, ask a couple of other hobbyst / engineers for any last pieces of advice online, and start creating a layout for the Printed Circuit Board (PCB).

As one of the last things I added also a goldpin header connected to two unused I/O pins of Atmega328 along with +5 V & GND connections. These are marked as Dev Ports as they are intendend to be used for future development purposes or external devices (like ESP32 with Wi-Fi capabilities) that may be connected to the board.

After making sure that all systems and hardware requirements should be met by the board and running final ERC in schematic and DRC in board file I exported gerber files for manufacturing. I ordered 5 test boards from JLC PCB, as their service was quite cheep and a good fit for the first batch of test boards. Below you can find some photos of the boards, soldering components in and first tries of connecting the charger to power supply.

*PCB related photos*

Software – from first sketches to the board library

I didn’t have much experience with “pure” C/C++ embedded software but I have been using the Arduino library for a long time, so I decided to stick to that option in this project as well (at least for the prototyping and first boards). I started with separate sketches for each functionality, and I tried to test them one by one before merging them into bigger files. Finally I decided to try my hand at Object Oriented Programming (OOP) and create a C++ library based on the Arduino library that would make the code more reliable, easy to read and edit.

**Taking voltage and current measurements**

…

**Setting output voltage (Proportional controller)**

…

**MPPT algorithm**

…

1. Callibration and testing
2. Conclusion and future development plans
3. Refrences