ISSN 2229-5518

# Automatic Maximum Power Point Tracking for Photovoltaic Generators using ARM Processor

Daphni S, Kumaresan N, Prabakaran C R

**Abstract**— This paper presents a fast and unconditionally stable maximum power point tracking scheme with high tracking efficiency is proposed for photovoltaic generators. Due to the variation of atmospheric conditions, the photovoltaic (PV) cells do not supply the stable maximum power usually. Some days PV cells provide low power, at that situation the low power can be boosted up to maximum power and applied for special applications.The MPPT power stage is implemented by means of a DC-DC converter and at the front end, most commonly pulse width modulation (PWM) is used. The processing analysis is controlled by ARM-11 (Raspberry pi), and the power result is viewed by any monitor display (T.V or laptop). Thus the system gives the stable maximum power for any situations even for supplying low power from photovoltaic generators. The system uses the ARM11 (Advanced Risc Machine) processor with an RTOS. The Real-Time Linux (RT Linux) operating system is used in this maximum power point tracking method. This paper work contains the RTLinux installation, kernel patching, configuration, compilation, and the maximum power point tracking.

**Index Terms**— Photovoltaic generators,Maximum power point tracking,Pulse width modulation,Advanced Risc Machine-11,RTOS, RT Linux, kernel patching

---

## 1 Introduction

A photovoltaic system is an arrangement of components designed to supply usable electric power for a variety of purposes, using the Sun as the power source. A photovoltaic array consists of multiple photovoltaic modules, casually referred to as solar panels, to convert solar radiation (sunlight) into usable direct current (DC) electricity. A photovoltaic system for residential, commercial, or industrial energy supply normally contains an array of photovoltaic (PV) modules, one or more DC to alternating current (AC) power converters (also known as inverters), a tracking system that supports the solar modules, electrical wiring and interconnections, and mounting for other components.

Optionally, a photovoltaic system may include any or all of the following: renewable energy credit revenue-grade meter, maximum power point tracker (MPPT), battery system and charger, GPS solar tracker, solar concentrators, solar irradiance sensors, anemometer, or task-specific accessories designed to meet specialized requirements for a system owner. The simplest PV systems power many of the small calculators and wrist watches we use every daySolar energy is radiant energy from the sun. It is vital to us because it provides the world-directly or indirectly with almost all of its energy.


- *Daphni S is currently pursuing masters degree program in Applied Electronics in Anna University Regional Centre Coimbatore, India, PH-8675857541. E-mail: daphnisamraj@gmail.com*
- *Dr.N.Kumaresan is currently working as Assistant Professor in Anna University Regional Centre Coimbatore, India, PH-9443484396. E-mail: prakumpriniv@gmail.com*
- *Mr.C.R.Prabakarn is currently pursuing masters degree program in Embedded System Technologies in Anna University Regional Centre Coimbatore, India, PH-9994023624. E-mail: praba14est@gmail.com*


In addition to providing the energy that sustains the world, solar energy is stored in fossil fuels and biomass, and is responsible for powering the water cycle and producing wind. Every day the sun radiates, or sends out, an enormous amount of energy. The sun radiates more energy in one second than people have used since the beginning of time! Solar energy comes from within the sun itself. Like other stars, the sun is a big ball of gases-mostly hydrogen and helium. The hydrogen atoms in the sun's core combine to form helium and radiant energy in a process called nuclear fusion. The conversion efficiency of a PV cell is the proportion of radiant energy the cell converts into electrical energy relative to the amount of radiant energy that is available and striking the PV cell. This is very important when discussing PV devices, because improving this efficiency is vital to making PV energy competitive with more traditional sources of energy.

An ARM processor is any of several 32-bit RISC (reduced instruction set computer) microprocessor s developed by Advanced RISC Machines, Ltd. The ARM architecture was originally conceived by Acorn Computers Ltd. in the 1980s. Since then, it has evolved into a family of microprocessors extensively used in consumer electronic devices such as mobile phone s, multimedia players, pocket calculator s and PDAs.

### 1.1 Motivation of the work

At given atmospheric conditions (mainly dependent on temperature [1] and insolation level), photovoltaic (PV) cells supply maximum power at a particular operating point—the maximum power point (MPP). Unlike conventional (fueled) power sources, it is desirable to operate PV systems at their MPP [1]–[13]. However, the MPP locus varies over a wide intensity [1], [2]. Instantaneous shading conditions and ageing of PV cells also affect the MPP locus. Hence, in order to achieve operation at the MPP, a time varying matching network which interfaces the varying source and the potentially varying load is required. The role of this matching network, called the maximum power point tracking network (MPPT), is to ensure operation of the PV generator (PVG) at its MPP, in the face of changing atmospheric conditions and load variations. Typically, the MPPT power stage is implemented by means of a dc-dc converter at the front end, most commonly pulse width modulated (PWM). In the case of standalone PVG this PWM controlled DC-DC converter may constitute the entire power stage, whereas in the case of a grid connected PVG, it would be followed by a dc-link capacitor and an inverter.

## 2 LRITERATURE SURVEY

Tsai-Fu Wu, et al., (2013) focused on integration and operation of a single-phase bidirectional inverter with two buck/boost maximum power point trackers (MPPTs) for dc-distribution applications. In a dc-distribution system, a bidirectional inverter is required to control the power flow between dc bus and ac grid, and to regulate the dc bus to a certain range of voltages.

Jungmoon Kim, and Chulwoo Kim,(2013) proposed a dc–dc boost converter with the maximum power point tracking (MPPT) technique for thermoelectric energy harvesting applications. The technique realizes variation tolerance by adjusting the switching frequency of the converter. A finely controlled zero-current switching (ZCS) scheme together with the accurate MPPT technique enhances the overall efficiency of the converter because of an optimal turn-on time generated by a one-shot pulse generator that is proposed. Moreover, the ZCS technique can deal with low- and high-temperature differences applied to the thermoelectric generator.

Guan-Chyun, et al., (2013) presented a two-phased tracking that forms a photovoltaic (PV) power-increment-aided incremental conductance (PI-INC) maximum power point tracking (MPPT) to improve the tracking behavior of the conventional INC MPPT. Enrico Bianconi, et al., (2013) introduced a novel maximum power point tracking (MPPT) technique aimed at maximizing the power produced by photovoltaic (PV) systems. The largest parts of the maximum power point tracking approaches presented in the literature are based on the sensing of the Photovoltaic generator voltage.

## 3 MPPT USING PWM AND BOOST CONVERTER

At given atmospheric conditions (mainly dependent on temperature and insolation level), photovoltaic (PV) cells supply maximum power at a particular operating Point-the maximum power point (MPP).The role of this matching network, called the maximum power point tracking network (MPPT), is to ensure operation of the PV generator (PVG) at its MPP, in the face of changing atmospheric conditions and load variations. Typically, the MPPT power stage is implemented by means of a dc-dc converter at the front end, most commonly pulse width modulated (PWM).

Various algorithms may perform MPPT. Important factors to consider when choosing a technique to perform MPPT are the ability of an algorithm to detect multiple maxima, costs, and convergence speed. The irradiance levels at different points on a solar panel's surface tend to vary. This variation leads to multiple local maxima power points in one system. The efficiency and complexity of an algorithm determine if the true maximum power point or a local maximum power point is calculated. In the latter case, the maximum electrical power is not extracted from the solar panel. The type of hardware used to monitor and control the MPPT system affect the cost of implementing it. The type of algorithm used largely determines the resources required to build an MPPT system.

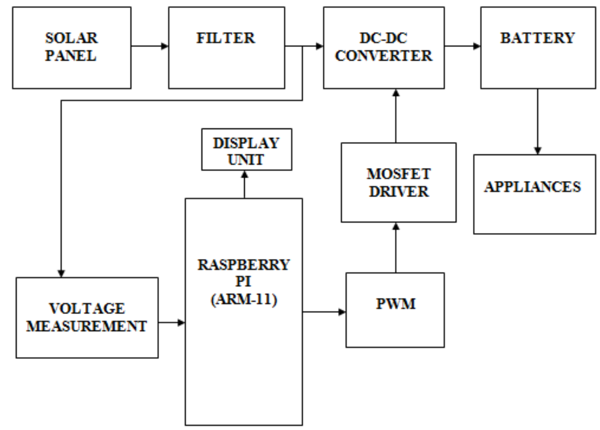


Fig. 1 Represents the block diagram of the proposed system. The block diagram consists of controller section, Boost converter, Maximum power point tracking section and power supply section

## 4 SYSTEM OVERVIEW

The system comprises some of the components like Solar Panel, Filter, Voltage Measurement, ARM-11 Processor, PWM, MOSFET Driver, DC-DC Converter, Battery, Appliances. ARM-11 Processor named as Raspberry Pi which controls all the processing of this work and displays the measurement values. This System consists of several additional interface components. The main components are

- Solar Panel
- Raspberry Pi
- PWM with MOSFET Driver
- Boost Converter
- Power Supplies

### 4.1 Solar Panel

A solar panel is a set of solar photovoltaic modules electrically connected and mounted on a supporting structure. A photovoltaic module is a packaged, connected assembly of solar cells. The solar module can be used as a component of a larger photovoltaic system to generate and supply electricity in commercial and residential applications. Each module is rated by its DC output power under standard test conditions (STC), and typically ranges from 100 to 320 watts. The efficiency of a module determines the area of a module given the same rated output - an 8% efficient 230 watt module will have twice the area of a 16% efficient 230 watt module.

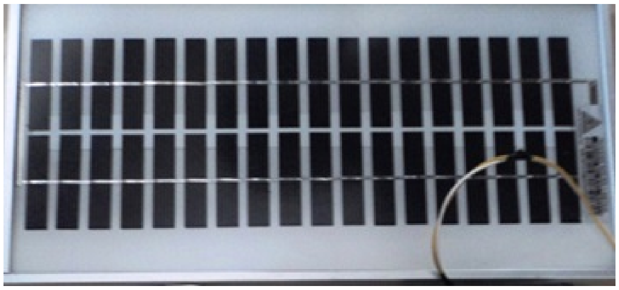

Fig.2.solar panel used for the hardware design module

### 4.2 Raspberry Pi

The ARM-11 Processor family provides the engine that powers many smart phones in production today and is also widely used in consumer, home and embedded applications. It delivers low power and a range of performance from 350 MHz in small area designs up to 1 GHz in speed-optimized designs in 45 and 65 nm. ARM-11 Processor software is compatible with all previous generations of ARM Processors, and introduces 32-bit SIMD for media processing, physically tagged catches to improve OS context switch performance, trust zone for hardware-enforced security, and tightly coupled memories for real-time applications.

The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming. The below Figure 3.3 shows the model of Raspberry Pi. The **Raspberry Pi** is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.
Broadcom's

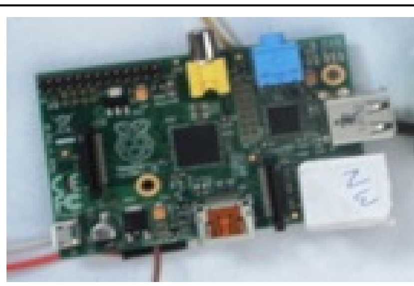

Fig.3. The ARM11 processor BCM2835 Raspberry Pi board used for the proposed system

BCM2835 combines the functionality that is currently required by up to five separate integrated devices, and is engineered with a dual-core CPU that optimizes system security while retaining the CPU's high-performance resources. The BCM2835-SoC also integrates a quad transcoder to enable consumers to stream data services and IP video to a range of devices. BCM2835 contains the following peripherals which may safely be accessed by the ARM:

- Timers, Interrupt controller
- GPIO, USB, PCM / I2S
- DMA controller
- I2C master, I2C / SPI slave
- SPI0, SPI1, SPI2
- PWM, UART0, UART1

## 5 System Design And Implementation

While Integrated Development Environments for standard Linux, such as the multi-language Eclipse and J Builder for the Java world are becoming more common, the usual development environment is the X11 windowing system in which the user develops code with the GNU tool chain featuring the GCC compiler. As for standard Linux, the common mode for real-time Linux code developers is to use the GNU tool chain along with their favorite editor to create and test code. The implementation language will be C to fit with the philosophy of the standard and real-time kernels. The real-time tasks are written as loadable kernel modules that can be inserted or removed from kernel space by user-space commands. Thus the real-time code can be developed and tested without rebooting the machine hosting the development and execution.

### 5.1 Kernal Space Tasks

Real-time tasks that are created to explicitly run in kernel space, typically as a loadable kernel module, suffer some restrictions on what software resources they may access. RT Linux and RTAI support this style of real-time execution. Standard Linux device drivers may not be called, so access to resources such as serial ports, Ethernet cards and other devices with entries in the /dev directory cannot be made with the Unix open() call. There are replacement drivers specifically written to work with real-time Linux available for most resources, including the serial and Ethernet devices just mentioned, as well as many devices common to industrial automation or scientific computing such as CAN Bus and data acquisition boards. System kernel resources also may not be used directly.

These include disk files normally accessed with fopen (), heap memory allocation and deallocation with malloc ()/free () and new/delete, console printing via printf() and traditional Unix inter process communication mechanisms such as pipes, shared memory, semaphores and message queues. However, real time Linux provides substitutes for many of these. Printf () is implemented using simple replacements like rtl_printf () (RTLinux) or rt_printk () (RTAI) that direct output to system log files such as /var/log/messages. Shared memory and first-in-first-out (FIFO) queues are supported through similar replacements, and allow real-time processes to communicate directly with non-real-time Linux processes. Other IPC features, such as semaphores and message queues, are supported for synchronization and communication between real-time tasks, but not between real-time tasks and non-real-time tasks.

### 5.2 RT Linux Porting

It describes that how to get-up-and running with prebuilt images of the u-boot; LINUX kernel and root file system.

U-boot.bin – The Universal Boot loader,
UImage – The LINUX kernel image.

ISSN 2229-5518

rootfs.jffs2 – A JFFS2 formatted root file system to be stored in NAND flash.

Besides the images a USB memory stick also needed and an MMC/SD card (If U-boot load from a MMC/SD card) in order to follow the steps below. A terminal application is also required as an interface towards the board.

1. Load the u-boot.bin file to the root directory in a FAT formatted MMC/SD card.
2. Load the uimage and rootfs.jffs files to the root directory of a USB memory stick.
3. Insert the MMC/SD card in the MMC/SD card slot on the QVGA Base board.
4. Insert the USB memory stick in the USB A connector.
5. Connect the USB with the board to the USB mini-B connector marked UART#0.
6. Install necessary FTDI USB drivers and identify which COM port that was assigned to the board.
7. Start the terminal application and connect to the COM port associated with the board.
8. Reset the board. Then the S1L boot loader will boot.

Load the root file system (rootfs.jffs2) from the USB memory stick and store it in NAND flash. Then load the LINUX kernel (UImage) from the USB memory stick and boot the kernel.

- The default u-boot environment has been prepared with a variable named update_fs.
- This will load the root file system and store it in NAND flash.
- When the root file system has been stored in NAND Flash it is time to load and boot the LINUX kernel.
- A variable named mtd boot is available in the default environment. This variable will setup the boot arguments to use a root file system in NAND flash (in an MTD partition), load the kernel and then boot it.

LINUX is now up-and-running. Load the root file system (rootfs.jffs2) from the USB memory stick and store it in NAND flash. Then load the LINUX kernel (UImage) from the USB memory stick and boot the kernel.

After porting Real Time LINUX into a target board (ARM processor) user can start write and execute application code in it. If the code is written in windows environment user can use various embedded development tools like IAR Workbench, Keil IDE etc,. If the code is written in LINUX environment user should use or write specific embedded tool for code development and download process. In this system the application is developed using Embedded C language. It is most suitable in many embedded real time applications.

## 6 Experimental Results

Figure 4 to 9 shows the hardware experimental setup of the system

### 6.1 Software Module Configuration

The following figure shows the puTTY and VNC Viewer Configuration also gives the startup conditions. VNC is platform-independent – a VNC viewer on one operating system may connect to a VNC server on the same or any other operating system.

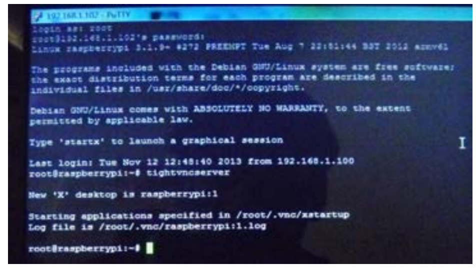

Fig.4. Initialization of RT Linux booted in ARM11 board

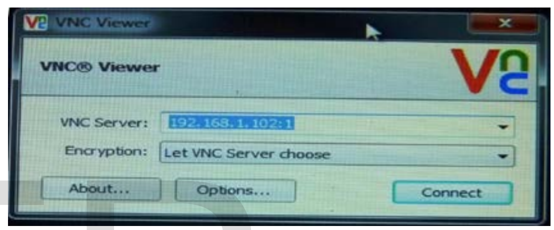


Fig.5. VNC server configuration for creating the virtual network

PuTTY is a free and open-sourceterminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet and rlogin. The name "PuTTY" has no definitive meaning, though "tty" is the name for a terminal in the Unixtradition, usually held to be short for Teletype. PuTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems.

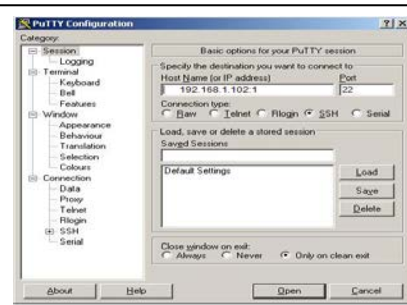


Fig.6. PuTTY configuration for secured shell host.

### 6.2 Hardware Setup

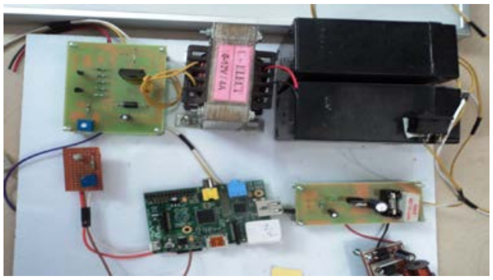

Fig.7. The hardware Setup consist of Processor, Battery storage and Boost converter

The above figures show the hardware module and total hardware setup with Raspberry Pi.

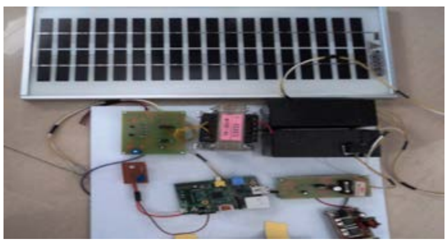

Fig.8. The full hardware setup of the proposed system

Maximum Tracking results from solar panel. The following figure 5.6 shows the Solar panel intensity and Battery voltage levels on the display.

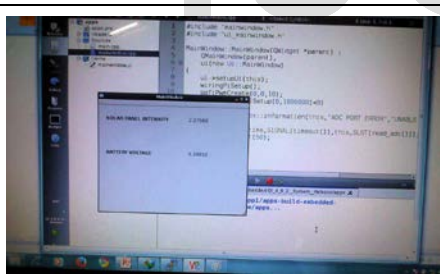

Fig.9. Representation of the output window in QT creator. The output window shows the maximum power point and the boosted power value

### Conclusion

In this project, Maximum Power Point Tracking from the Photovoltaic Generators using PWM, MOSFET Driver circuit and DC-DC Converter has been designed and implemented. Here the controller used as ARM-11 (Raspberry Pi). The controller controls all the operation and display the tracking values on the display unit. The voltage from the Solar Panel can be filtered by the filter and the low voltage can be given to the boost-up circuit. The Low voltage can be increased by boost-up circuit and stored in the batteries for further applications.

## REFERENCES


[1] Bianconi, E, Calvente, J, Giral, R, Petrone, G, Ramos-Paja, C.A, Spagnuolo, G and Vitelli, M,"A fast current-based MPPT technique based on sliding mode control", in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, pp. 59–64, 2011.

[2] Cabal, C, Alonso, C, Cid-Pastor, A, Estibals, B, Seguier, L, Leyva, R, Schweitz, G and Alzieu J, "Adaptive digital MPPT control for photovoltaic applications", *in Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, pp. 2414–2419, 2007.

[3] Coelho, R.F, Concer, F.M and Martins, D.C, "Analytical and experimental analysis of DC-DC converters in photovoltaic maximum power point tracking applications", in *Proc. 36th Annu. Conf. IEEE Ind. Electron.Soc. (IECON)*, pp. 2778–2783, 2010.

[4] Gao, L, Dougal, R.A, Liu, S and Iotova, A, "Parallel-Connected solar PV system to address partial and rapidly fluctuating shadow conditions", *IEEE Trans. Ind. Electron*, vol. 56, no. 5, pp. 1548–1556, 2009.

[5] Hussein, K.H, Muta, I, Hoshino, T and Osakada, M, "Maximum photovoltaic power tracking: An algorithm for rapidly changing atmospheric conditions", in *IEEE Proc. Gener., Transm, Distrib.*,vol. 142, pp. 59–64,1995..

[6] Jain, S and Agarwal, V, "A new algorithm for rapid tracking of approximate maximum power point in photovoltaic systems", *IEEE Power Electron. Lett*, vol.2, no. 1, pp. 16–19, 2004.

[7] Leyva, R, Alonso, C, Queinnec, I, Cid-Pastor, A, Lagrange, D. and Martinez Salamero, L, "MPPT of photovoltaic systems using extremum-Seeking control", *IEEE Trans. Aerosp. Electron. Syst.*, vol.42, pp. 249–258, 2006.

[8] Leppaaho, J and Suntio, T, "Dynamic properties of PCM-controlled current-fed boost converter in photovoltaic system interfacing", in *Proc. 14th Eur. Conf. Power Electron. Appl. (EPE)*, pp.1–10, 2011.

[9] Martinez-Salamero, L. et al, "Why is sliding mode control methodology needed for power converters", *in Proc. 14th Int. Power Electron. Motion Control Conf. (EPE/PEMC)*, pp. S9-25–S9-31, 2010.

[10] Scarpa, V.V, Spiazzi, G. and Buso, S, "Low complexity MPPT technique exploiting the effect of the PV cell series resistance", *IEEE Trans. Ind. Electron*, vol. 56, no. 5, pp. 1531–1538, 2008.

[11] Sera, D, Teodorescu, R, Hantschel, J and Knoll, M, "Optimized maximum power point tracker for fast-changing environmental conditions", *IEEE Trans. Ind. Electron*, vol. 55, pp. 2629–2637, 2008.

[12] Shmilovitz, D, "On the control of photovoltaic maximum power point tracker via output parameters", *IEE Trans. Power Appl*, vol. 152, no.2, pp. 239-248, 2005.

[13] Sokolov, M and Shmilovitz, D, "A modified MPPT scheme for accelerated Convergence", *IEEE Trans. Energy Convers*, vol. 23, pp.1105–1107, 2008.