Practical Guide to Implementing Solar Panel MPPT Algorithms

Authors: Mihnea Rosu-Hamzescu
Sergiu Oprea
Microchip Technology Inc.

INTRODUCTION

Using a solar panel or an array of panels without a controller that can perform Maximum Power Point Tracking (MPPT) will often result in wasted power, which ultimately results in the need to install more panels for the same power requirement. For smaller/cheaper devices that have the battery connected directly to the panel, this will also result in premature battery failure or capacity loss, due to the lack of a proper end-of-charge procedure and higher voltage. In the short term, not using an MPPT controller will result in a higher installation cost and, in time, the costs will escalate due to eventual equipment failure. Even with a proper charge controller, the prospect of having to pay 30-50% more up front for additional solar panels makes the MPPT controller very attractive.

This application note describes how to implement MPPT using the most popular switching power supply topologies. There are many published works on this topic, but only a tiny portion of them show how to actually implement the algorithms in hardware, as well as state common problems and pitfalls. Even when using the simplest MPPT algorithm with a well-designed synchronous switching power supply, it can be expected that at least 90% of the panel's available power will end up in the battery, so the benefits are obvious.

The topology presented in this application note is an inverse SEPIC, but the techniques used here can be applied to buck, boost and SEPIC converters. The buck converter is a special case, since it has a linear voltage transfer function when operating in Continuous Conduction Mode (CCM). This simplifies things a lot, and the MPPT controller can be implemented by operating directly on the converter duty cycle. The other topologies have a nonlinear voltage transfer function, and operating directly on the duty cycle will yield unpredictable results, especially at high duty cycles. In this case, the algorithm modifies the solar panel operating voltage by using a proportional integral (PI) control loop, which steers the voltage to the desired value.

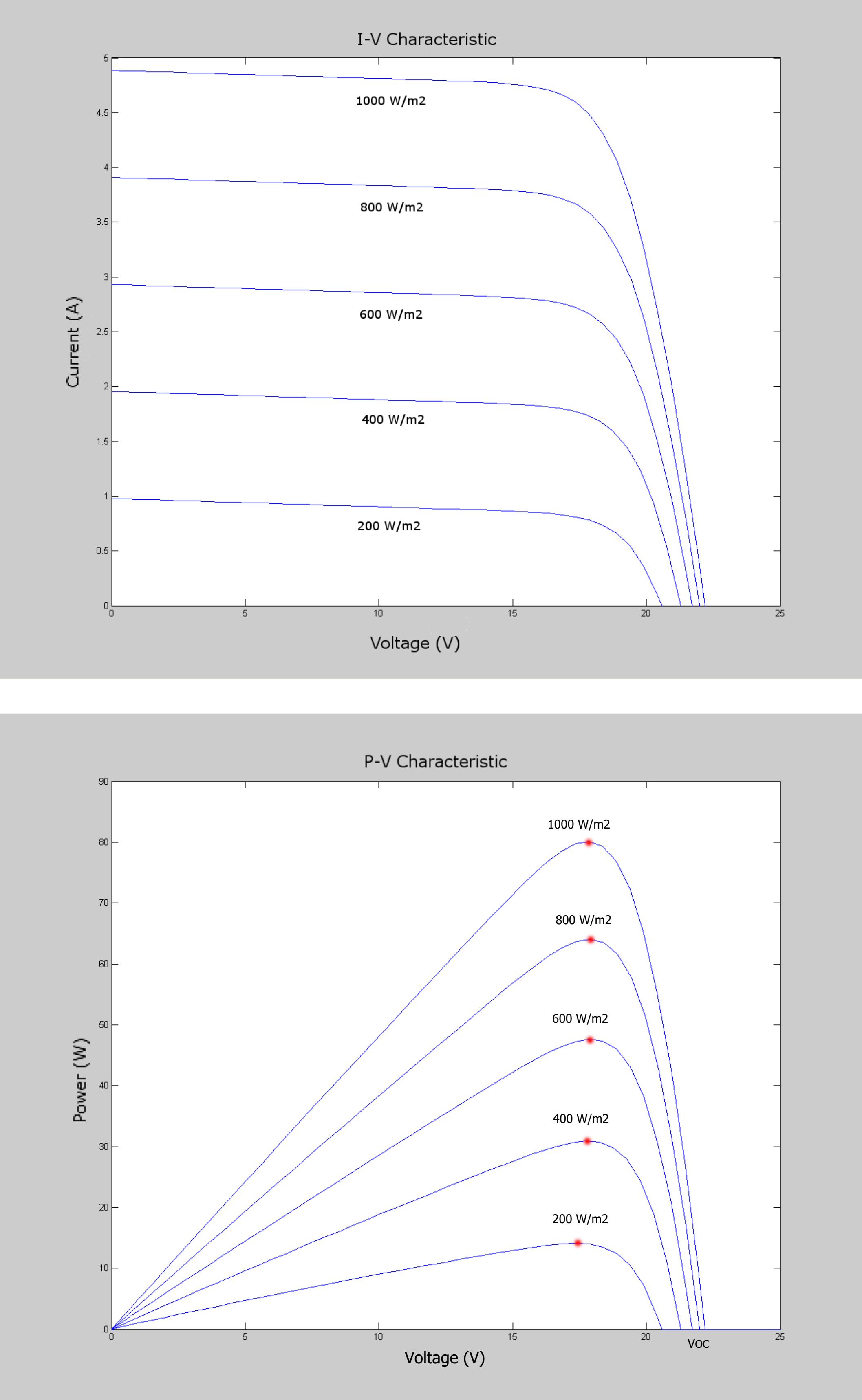
SOLAR PANEL MPPT

The main problem solved by the MPPT algorithms is to automatically find the panel operating voltage that allows maximum power output. In a larger system, connecting a single MPPT controller to multiple panels will yield good results, but, in the case of partial shading, the combined power output graph will have multiple peaks and valleys (local maxima). This will confuse most MPPT algorithms and make them track incorrectly. Some techniques to solve problems related to partial shading have been proposed, but they either need to use additional equipment (like extra monitoring cells, extra switches and current sensors for sweeping panel current), or complicated models based on the panel characteristics (panel array dependent). These techniques only make sense in large solar panel installations, and are not within the scope of this application note.

Ideally, each panel or small cluster of panels should have their own MPPT controller. This way the risk of partial shading is minimized, each panel is allowed to function at peak efficiency, and the design problems related to converters handling more than 20-30A are eliminated.

A typical solar panel power graph (Figure 1) shows the open circuit voltage to the right of the maximum power point. The open circuit voltage (V_{OC}) is obviously the maximum voltage that the panel outputs, but no power is drawn. The short-circuit current of the panel (I_{SC}) is another important parameter, because it is the absolute maximum current you can get from the panel.

FIGURE 1: SOLAR PANEL CHARACTERISTICS



The literature on this subject generally agrees that the maximum amount of power that can be extracted from a panel depends on three important factors: irradiance, temperature and load.

Matching panel and load impedances with a DC-DC converter makes sense, because for example, if you have a 5V/2A load, and a 20W panel that has the MPP at 17.5V/1.15A, connecting the load directly will not work. Considering a simple resistive load, and the short-circuit current of 1.25A, the panel will only be able to provide about 3V/1.2A, or less than 4W out of 20W.

Temperature mainly changes the panel voltage operating point, while irradiance mainly changes the panel operating current. Figure 1 shows the effect of different irradiance levels on the panel voltage, current and power.

There are a few MPPT algorithms that can be easily implemented using an 8-bit microcontroller.

FRACTIONAL OPEN CIRCUIT VOLTAGE

The maximum power point voltage has a linear dependency on the open circuit voltage V_{OC} under different irradiance and temperature conditions. Computing the MPP (Maximum Power Point) comes down to:

EQUATION 1:

$$V_{MPP} = k_V V_{OC},$$

The constant k depends on the type and configuration of the photovoltaic panel. The open circuit voltage must be measured and the MPP determined in some way for different ambient conditions. Usually, the system disconnects the load periodically to measure V_{OC} and calculate the operating voltage. This method has some clear disadvantages, temporary loss of power being an obvious one. An alternate method would be to use one or more monitoring cells, but they also need to be chosen and placed very carefully to reflect the true open circuit voltage of the system.

Although this method is quite simple and robust and does not require a microcontroller, the constant only allows a crude approximation of the MPP. Other algorithms will significantly increase the top power drawn from the same PV installation.

FRACTIONAL SHORT CIRCUIT CURRENT

The MPP can also be determined from the short-circuit current of the panel (I_{SC}), because I_{MPP} is linearly related to it under varying atmospheric conditions.

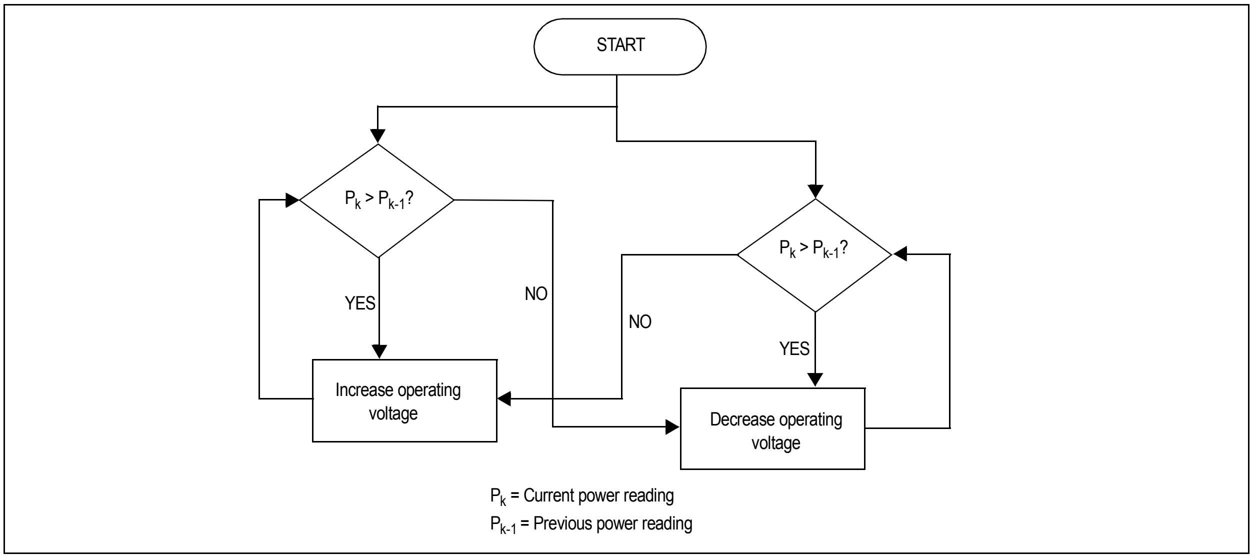
EQUATION 2:

$$I_{MPP} = k_I I_{SC}$$

Similar to fractional open circuit voltage, the constant must be determined for each type of system. Determining I_{SC} is more challenging, because doing so from time to time not only increases power loss and heat dissipation, but also requires an additional switch and current sensor. Obviously, this increases component count and cost. The simplest implementations do not require microcontrollers, but for better accuracy and to solve problems related to partial shading, more processing power is necessary to sweep the panel current from 0 to I_{SC}, and memorize the output voltage profile.

PERTURB AND OBSERVE (P&O)

FIGURE 2: P&O ALGORITHM



P&O is one of the most discussed and used algorithms for MPPT. The algorithm involves introducing a perturbation in the panel operating voltage. Modifying the panel voltage is done by modifying the converter duty cycle. The way this is done becomes important for some converter topologies.

Looking at Figure 2 makes it easy to understand that decreasing voltage on the right side of the MPP increases power. Also, increasing voltage on the left side of the MPP increases power. This is the main idea behind P&O.

Let’s say that, after performing an increase in the panel operating voltage, the algorithm compares the current power reading with the previous one. If the power has increased, it keeps the same direction (increase voltage), otherwise it changes direction (decrease voltage). This process is repeated at each MPP tracking step until the MPP is reached.

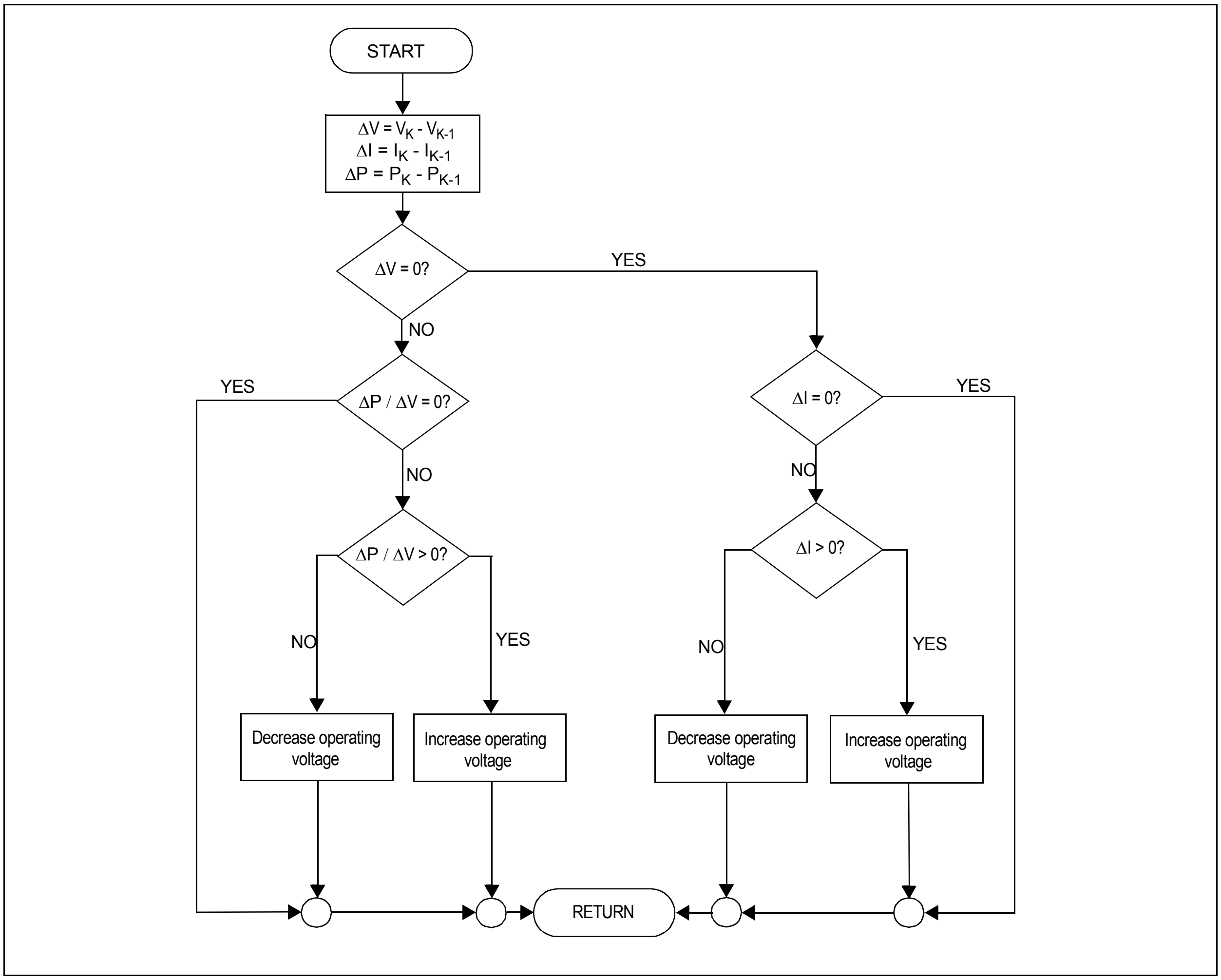
After reaching the MPP, the algorithm naturally oscillates around the correct value.

The basic algorithm uses a fixed step to increase or decrease voltage. The size of the step determines the size of the deviation while oscillating about the MPP. Having a smaller step will help reduce the oscillation, but will slow down tracking, while having a bigger step will help reach MPP faster, but will increase power loss when it oscillates.

To be able to implement P&O MPPT, the application needs to measure the panel voltage and current. While implementations that use only one sensor exist, they take advantage of certain hardware specifics, so a general purpose implementation will still need two sensors.

INCREMENTAL CONDUCTANCE

FIGURE 3: INCCOND ALGORITHM



The incremental conductance algorithm uses the fact that the panel power curve derivative (or slope) versus voltage is 0 at MPP, positive on the left side and negative on the right side of the MPP.

EQUATION 3:

$$(1)\begin{cases}\dfrac{dP}{dV} = 0,\ at\ MPP\\ \dfrac{dP}{dV} > 0,\ left\ of\ MPP\\ \dfrac{dP}{dV} < 0,\ right\ of\ MPP\end{cases}$$

The power derivative can be also written as:

EQUATION 4:

$$\frac{dP}{dV} = \frac{d(IV)}{dV} = \frac{IdV}{dV} + \frac{VdI}{dV} = I + V\frac{dI}{dV}$$

$$I + V\frac{dI}{dV} \cong I + V\frac{\Delta I}{\Delta V}$$

So the first bundle of equations (1) can be rewritten as:

EQUATION 5:

$$(2)\begin{cases}\dfrac{\Delta I}{\Delta V} = -\dfrac{I}{V},\ at\ MPP\\ \dfrac{\Delta I}{\Delta V} > -\dfrac{I}{V},\ left\ of\ MPP\\ \dfrac{\Delta I}{\Delta V} < -\dfrac{I}{V},\ right\ of\ MPP\end{cases}$$

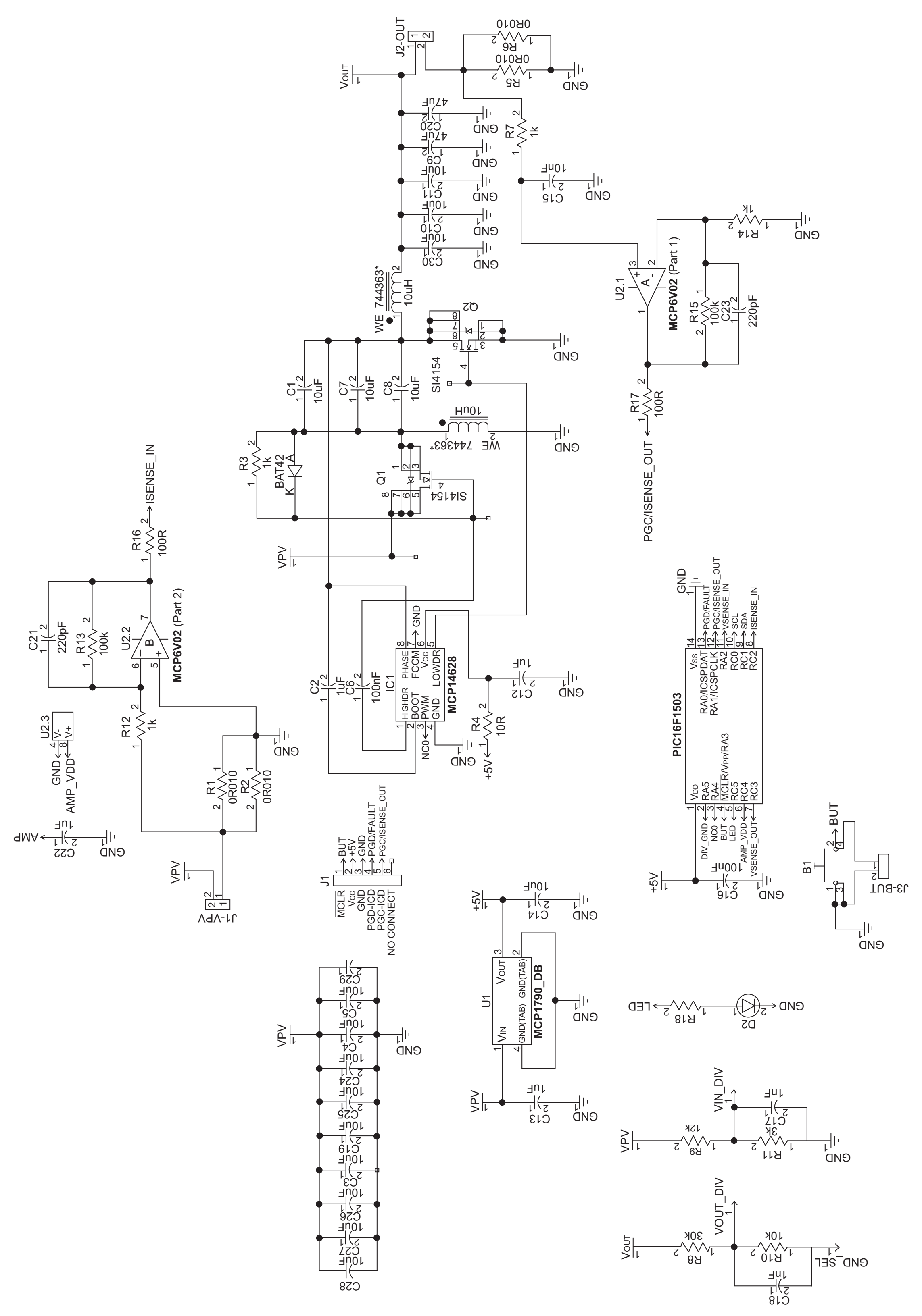
The main idea is to compare the incremental conductance ($\frac{\Delta I}{\Delta V}$) to the instantaneous conductance ($\frac{I}{V}$). Depending on the result, the panel operating voltage is either increased, or decreased until the MPP is reached. Unlike the P&O algorithm, which naturally oscillates around the MPP, incremental conductance stops modifying the operating voltage when the correct value is reached. A change in the panel current will restart the MPP tracking. Depending on the ambient conditions, the same functionality may be achieved by using the initial equation ($\frac{\Delta P}{\Delta V}$).

The basic incremental conductance algorithm uses a fixed step size for the panel operating voltage updates. Using a bigger step size will speed up tracking, but may also cause the algorithm to oscillate around the MPP instead of locking on.

Implementing the incremental conductance algorithm requires the voltage and the current output values from the panel (two sensors). Because it needs to keep track of previous voltage and current values, this algorithm is usually implemented using a PIC® device or a DSP.

MPPT HARDWARE PLATFORM

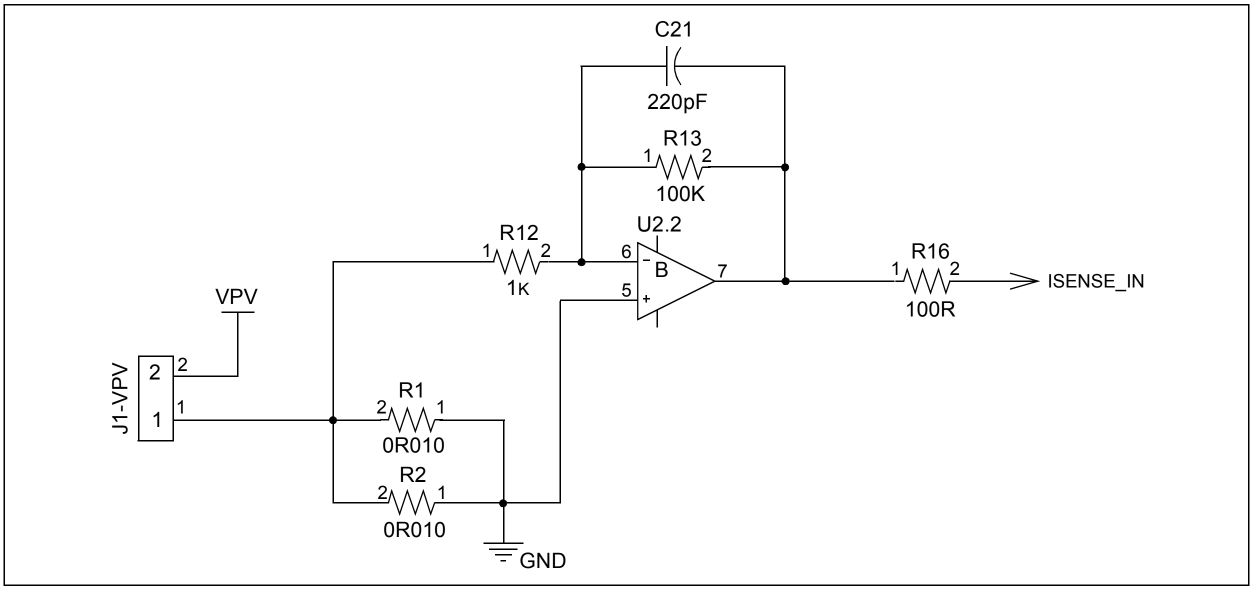
FIGURE 4: MPPT SCHEMATIC



For demonstrating and testing the MPPT algorithms in this application note, the synchronous inverse SEPIC (ZETA) hardware platform presented in AN1467, *"High-Power CC/CV Battery Charger Using an Inverse SEPIC (Zeta) Topology"*, was used. The NCO peripheral of the PIC16F1503 is used to generate a high resolution 15-bit fixed on-time PWM for the control scheme.

In general, the implementation is similar to a DC-DC converter with current and voltage sensors on the input side (solar panel). If battery charging is implemented on the same platform, then another set of current and voltage sensors is required on the output side.

FIGURE 5: PV CURRENT SENSOR

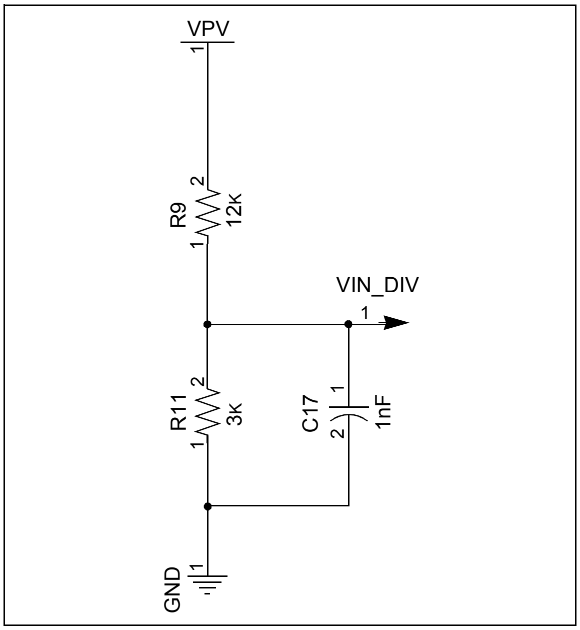


For best accuracy, a high-side sensor should be used, but this complicates things and requires a specialized chip. It is easier to use a low-side shunt and an inverting amplifier, but the microcontroller and amplifier current is added to the measurement. Microchip's MCP6V02 is perfect for this task due to the high amplification that is required to keep power dissipation at a minimum. This amplifier only has 2 uV of voltage offset. Even without any prior calibration, we can expect 5% of current measuring accuracy (all resistors 1%).

Since most of the switching noise is amplified along with the current shunt voltage, it is recommended to reduce the amplifier bandwidth. Because the prototype used for testing the algorithms relies on a proportional integral (PI) loop to regulate voltage and current, and the loop update rate is 1 kHz, the corner frequency is set around 7 kHz. Having a properly filtered signal is very important because algorithms like INCCOND are easily disturbed by noise and their performance degrades.

Normally, for the MPPT, it is desirable to have a very fast update rate and a small increment step. This way, tracking accuracy is improved (small increment), and the speed loss is countered by the high update rate. In practice, this is not always possible.

FIGURE 6: PV VOLTAGE SENSOR



The current shunt amplifier amplification, ADC resolution and noise will dictate the lower limit of the step size. This is why the current shunt amplification needs to be large enough, so that the output for the maximum allowed current will be close to the ADC input voltage limit. The test board was designed to work at 8A input current with a 5 mOhm shunt. Current shunt amplification is 100, so the maximum output value is:

EQUATION 6:

$$8A*0.005Ohm*100 = 4.0V$$

Please note that the maximum output value of the amplifier should allow enough headroom for the PI loop to quickly handle overcurrent conditions.

The ADC reference is 5V with 10 bits of resolution, making this a good choice. The current resolution is given by the ADC resolution and, in this case, is about 10 mA. When measuring, a few bits of noise is expected, so choose the MPPT step accordingly.

Regarding the panel voltage sensor, the divider ratio should be chosen depending on the panel open circuit voltage, which is the maximum panel voltage. On the testing board, the ADC reference is 5V and a 36-cell panel was used. This means V_{OC} should not go above 22V (36 * 0.6V). Also, one must think about maximizing voltage resolution, so a 1/5 divider was chosen, allowing a maximum input voltage of 25V. The voltage resolution depends on the divider ratio and the ADC resolution, in this case about 25 mV.

EQUATION 7:

$$\frac{5V}{1023}*5 = 0.00244V$$

The MPPT update rate is also limited by several factors. The most important thing is the PI loop update rate, which should be significantly faster than the MPPT update rate. Obviously, this is important because between MPPT updates the PI loop must have enough steps to be able to steer the panel voltage to the new value. The implementation used to test the algorithms runs the MPPT update at 40 Hz, while the PI loop runs at 1 kHz (25 times faster).

The board input capacitance will have an important contribution to the MPPT accuracy in low illumination conditions. While input capacitance is good for limiting current ripple, it will also slow down panel voltage variations. This is extremely important for tracking at low-power levels. Too little capacitance will cause stress to the components (high-voltage ripple – remember the inverse SEPIC has discontinuous input current), but too much capacitance will cause the panel voltage regulation loop in Tracking mode to be unable to reach the set point between MPPT updates.

MPPT SOFTWARE IMPLEMENTATION

The PIC MCU used for this prototype is an 8-bit device without a hardware multiplier, so the computational power is limited. This makes everything extremely attractive from a cost standpoint, but also makes the implementation more challenging.

Having a multi-step battery charger and an MPPT controller running on the same chip is a challenge due to the following issues:

- Device must regulate output current
- Device must regulate output voltage
- Device must track the panel MPP
- Device must run a battery-charging state machine
- Device has limited computational power and must run only one regulation loop

The algorithms presented in this application note track the maximum power point by modifying the panel operating voltage. This solution is not topology-dependent and is the best solution overall. Modifying the converter duty cycle directly only works well with the buck topology, which has a linear voltage transfer function.

THE MAIN PROGRAM LOOP

CODE EXAMPLE 1:

```
      Initialize_Hardware();
      while(1)
      {
      if(T0IF)
                 {
                         T0IF = 0;
                         if(but_cnt) but_cnt--;
                         if(track && mppt_calc) mppt_calc--;
                         if(second) second--;

                         read_ADC();

                         if(!track)
                         {
                                 if(battery_state != FAULT)
                                 {
                                         cc_cv_mode();
                                         if(!cmode) pid(vout, vref); else
                                                 pid(iout, iref);

                                         if(increment >= dmax)  track = TRACK_DELAY;
                                 }
                         } else
                         {
                                 if(mppt_calc < MPPT_AVERAGE)
                                 {
                                         f_vin += vin;
                                         f_iin += iin;
                                 }
                                 if(!mppt_calc)
                                 {
                                         mppt_calc = MPPT_INTERVAL;

                                         #ifdef MPPT_PO
                                                 mppt_PO();
                                         #endif

                                         #ifdef MPPT_INCCOND
                                                 mppt_INCCOND();
                                         #endif
                                 }
                                 pid(vinref, vin);

                                 if(vout > vref || iout > iref)
                                 {
                                         track--;
                                         dmax = increment;
                                         if(!track) Init_State_Machine();
                                 } else
                                         track = TRACK_DELAY;
                         }
                         if(!second)
                         {
                                 second = SECOND_COUNT;
                                 if(!track) Battery_State_Machine();
                         }
                 }
         }
}
```

The main program loop structure is similar to the one used in the CC/CV battery-charging code attached to AN1467. The MPPT tracking code is added to the basic output regulation code and battery-charging library.

This implementation uses the fact that the device can either track the panel MPP, or regulate the output (but not both of them at the same time). The variable **track** shows whether the device is tracking or regulating the output.

While tracking the panel MPP, a number of input voltage and current samples are summed together for noise reduction, and then fed to the selected MPPT algorithm. The MPPT algorithm modifies the input voltage reference, and the PI loop steers the panel operating voltage to that reference voltage. The PI control loop needs to run many times faster than the MPPT algorithm, so that the panel voltage has enough time to stabilize. Observe that the set point and feedback terms fed to the PI function are reversed, because increasing the duty cycle causes the panel voltage to decrease.

If the output voltage or current are over the set limits, the Tracking mode ends and the control loop starts regulating output. This means that the panel can provide more power than the battery (or load) can absorb. The converter duty cycle is memorized when the Tracking mode ends. Normally, the output voltage is fixed so when the panel voltage goes down, the duty cycle needs to be increased to maintain the output. If the duty cycle is increased above the memorized value, then the panel voltage has fallen below the MPP. This is a very simple and easy way of knowing that the main loop needs to return to the Tracking mode. Other methods could use output current or voltage as indicators. If in Regulation mode, but not able to reach the set voltage or current, it is clear that the panel is not able to provide enough power for the load, and the main loop needs to switch back to Tracking mode.

In systems which use batteries to store energy, it is unavoidable to waste energy, if the panels provide more power than the batteries and load can absorb. In grid tied inverter applications this is not an issue, as all the available energy is pumped into the power grid.

One other important thing is that the battery state machine will only update in output Regulation mode. In Tracking mode, neither the output voltage, nor current limit is reached, so the charge termination protocols will not function properly. Even in low light conditions (slow charging), if the battery reaches the constant voltage stage, charge will be terminated as soon as the current falls under the threshold. To avoid keeping the cell or battery at high voltage, which is known to cause damage in the long run, another termination condition can be added to terminate charge if the current is too low, but the voltage is over a certain threshold. This is mostly required to protect Li-Ion chemistry batteries, since lead-acid is much cheaper and more tolerant to abuse.

Important variables and defines used here:

- Variable “**track**” shows if the main loop is tracking MPP or regulating output. True means it is tracking. The define **`TRACK_DELAY`** is a debouncing value for switching between tracking and output regulation. While in Tracking mode, the output voltage or current needs to be over the set limits for a number of `TRACK_DELAY` main timer ticks before switching over.
- Counter “**`mppt_calc`**” is the number of main loop timer ticks between MPPT updates. The counter is initialized using the define **`MPPT_INTERVAL`** from *Hardware.h*.
- “**`MPPT_AVERAGE`**” defines the number of samples averaged for use in the MPPT function. Always the last `MPPT_AVERAGE` samples before the MPPT update are averaged together. This is important because the first few samples after the update may contain transient values, as the PI control loop tries to steer the panel voltage to the set value.
- The variables “**`f_vin`**” and “**`f_iin`**” hold the averaged input samples for input voltage and current, used in the MPPT function.
- Defines “**`MPPT_PO`**” or “**`MPPT_INCCOND`**” tell the main loop which of the MPP tracking algorithms to use. Perturb and observe and incremental conductance algorithms are available in this implementation.
- The variable “**`vinref`**” holds the current operating voltage set for the panel. The PI loop tries to steer the panel voltage towards this value.

PERTURB AND OBSERVE MPPT IMPLEMENTATION

CODE EXAMPLE 2:

```
void mppt_PO(void)
{
        power = (long) f_vin * (long) f_iin;
        if(power < l_power) updown ^= 1;
        if(!updown) vinref -= MPPT_STEP; else vinref += MPPT_STEP;

        l_power = power;
        f_vin = 0;
        f_iin = 0;
}
```

Implementing a P&O algorithm in software is quite simple and in most cases, the difficulty comes from correctly designing the hardware platform. The fun thing is that it can also be easily adapted to wind turbines.

The algorithm calculates the power drawn from the panel using the averaged readings of the input voltage (`f_vin`) and current (`f_iin`). The power value is memorized at each iteration and is compared to the calculated power. If power has decreased, the algorithm changes direction. The MPPT step is user definable.

For this algorithm, simplicity and robustness are the main advantages, but because the panel operating voltage naturally oscillates about the MPP, some of the available power is lost.

INCREMENTAL CONDUCTANCE MPPT IMPLEMENTATION

CODE EXAMPLE 3:

```
void mppt_INCCOND(void)
{
long delta_i;
long delta_v;
long delta_p;

    power = (long) f_iin * f_vin;
    delta_p = power - l_power;

    delta_i = (long) f_iin - fl_iin;
    delta_v = (long) f_vin - fl_vin;

    if(delta_v)
    {
        ineq = delta_p / delta_v;
        if(ineq > 0) vinref += MPPT_STEP; else
        if(ineq < 0) vinref -= MPPT_STEP;

    } else
    {
            if(delta_i > 0) vinref += MPPT_STEP; else
            if(delta_i < 0) vinref -= MPPT_STEP;
    }

    fl_iin = f_iin;
    fl_vin = f_vin;
    l_power = power;

    f_iin = 0;
    f_vin = 0;
}
```

The incremental conductance algorithm is a bit more complicated to implement but it has the advantage of locking on the MPP.

For the calculations, in addition to the last power value, it is required to memorize the last voltage and current value. The power (ΔP), current (ΔI) and voltage (ΔV) differences are calculated.

If using the second form of the incremental conductance equations, it is not required to calculate the panel power. The instantaneous and difference values for the current and voltage are used, but in this case, proved to yield poor results. Even with filtering, the current readings were noisy enough to cause the incremental conductance ($\frac{\Delta I}{\Delta V}$) to change sign erratically. This confused the algorithm and caused it to get stuck in many cases.

To avoid this issue, the initial form of the incremental conductance equation was selected, which uses power and voltage for the calculations. For negative power-voltage slopes, the panel operating voltage is decreased and, for positive slopes, the panel operating voltage is increased. Even if some extra calculations were done, this yielded adequate results and the algorithm tracked maximum power correctly.

When the voltage difference is zero, the slope is not calculated (zero denominator), and the algorithm stops tracking. If the current difference (ΔI) is not zero, tracking is resumed.

Some improvements can be added that will not allow the algorithm to oscillate about the MPP due to noise. For example, instead of using the panel voltage readings in the calculations, the set reference value can be used. If the PI loop works correctly, this should be the operating voltage. This modification, together with a form of dead-band for the power-voltage slope value and current difference value, will ensure locking on the MPP. Only variations of the input current with a modulus greater than a preset value will cause the algorithm to resume tracking.

CONCLUSIONS

Using MPPT with solar panel installations has clear advantages. The initial investment is smaller because smaller panel wattage is required (very little potential power is wasted), and adding correct battery-charging algorithms will also decrease operating costs (batteries are protected and last longer).

MPPT algorithms are simple enough, but implementing a working MPPT controller is not a simple task, because it is required to know the particularities of the underlying switching converter. Many of the so-called scientific papers published on this topic simply use computer simulations instead of real hardware, and the readers find themselves lacking vital information.

In particular, the coexistence of the MPPT algorithm, the output regulation loop and the battery-charging state machine is a very interesting topic, which is often overlooked. By carefully defining the transition conditions between tracking and regulating, one PI-type loop is used to perform either MPPT or output regulation. The battery state machine is less processor-intensive and runs in parallel with the output regulation loop. This way the whole control algorithm, plus the battery-charging library, runs on a 4 MIPS® 8-bit PIC MCU with 2 kWords of Flash.

This application note presents two MPPT algorithms implemented on a synchronous inverse SEPIC converter. The same code can be used with minimal tuning on other topologies like buck, boost or SEPIC.

By utilizing the techniques presented in this application note, it is possible to optimize the cost and extend the life of any solar powered application ranging from a few watts to two hundred watts by adding MPPT.

REFERENCES

AN1467, High-Power CC/CV Battery Charger Using an Inverse SEPIC (Zeta) Topology (DS01467):

http://ww1.microchip.com/downloads/en/AppNotes/01467A.pdf

PIC16(L)F1503 14-Pin Flash, 8-Bit MCU Data Sheet (DS41607):

http://ww1.microchip.com/downloads/en/DeviceDoc/41607A.pdf

2A Synchronous Buck Power MOSFET Driver (DS22083):

http://ww1.microchip.com/downloads/en/DeviceDoc/22083a.pdf

MCP6V01/2/3 300µA, Auto-Zeroed Op Amps Data Sheet (DS22058)*:*

http://ww1.microchip.com/downloads/en/DeviceDoc/22058c.pdf

MCP1790/MCP1791 70mA, High Voltage Regulator Data Sheet (DS22075)*:*

http://ww1.microchip.com/downloads/en/DeviceDoc/22075b.pdf

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

ISBN: 9781620772164

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12