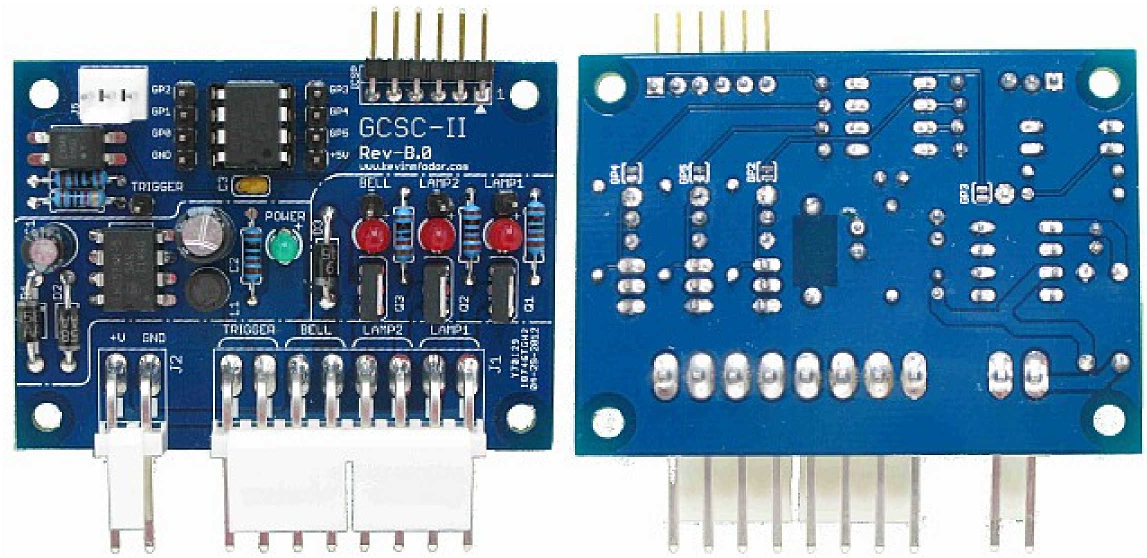
Grade Crossing Signal Controller II

Revision B.0

Requirements, Design and Implementation

Written By: Kevin Fodor

www.kevinmfodor.com



Background

Some time ago there has been some interest in my Railroad Signal Crossing Project posted on my project web site. One individual intends to build his own barrel-car train which will be able to pull around kids, which should prove to be a huge amount of fun for them. Being of a train theme he was interested in constructing four grade crossing signals (cross bucks) presumably to lay about the area where the barrel cars would follow. Seeing my project on my website he contacted me for more detailed plans on the construction of the signal post itself as well as the electronic controller for the lights and bell.

The physical construction went well and he made a great deal of improvements to make it easier to construct the signal itself. The electronics however were a bit more challenging. My original design was cobbled or hacked together as a work-in-progress. This means I didn't set out to design anything. Just kind of put enough components together (wire-wrapped) and debugged the circuit till it worked. Consequently other than the schematic which documents the circuit, there is very little information on how to actually duplicate the circuit (wiring points, etc.). Of course my intent was never to make more than one one them anyway. Now that there is interest in making more than one, a better plan and approach is required.

Purpose

This document will serve as a collection of objectives, requirements and design notes accumulated overtime to guide the re-design of this controller.

Current Design Weaknesses

The re-design is aimed at making the controller simpler to build and more reliable among many other things. To address what improvements are to be made, below are the weaknesses I see in the existing controller design;

- The existing design is wire-wrapped onto proto-board. This makes it very difficult to debug, copy and maintain as wires are fragile and break easily. Additionally, connection points are not easily observed.
- Overall the part count is rather high for what should be a very simple circuit.
- The existing design uses a 556 (dual 555) timer to both trigger the signal (and control the on time) as well as the astable operation of the lights and bell. These devices are notoriously problematic and I have found often false trigger if great care is not observed. Additionally to obtain a square wave pulse (50% duty cycle) a diode is required, adding to the overall part count.
- The power output for the lights and bell are based on power transistors (TIP31 and TIP120). Although they appear to do the job, they are not very efficient and require drive transistors to operate them. They are also more difficult to use and generate significant heat while operating.
- The bell signal is derived from light flashing signal (derivative of the square wave using a diode, D2 and D3). This adds to the number of component required and complexity. More importantly however this also makes timing the bell “strike” time somewhat tricky as it is subject to the RC circuits (C9, R10) and (C10, R11). Along with the drive circuits, these parts are duplicated for each clock edge to trigger the bell.
- Separate transformers were needed since the bell being an inductive load tended to drop the supply voltage to the timer and other circuitry causing all sorts of “drop-out” problems. Transformers are expensive and it would be nice to require only one, and only if being powered by AC household line current. Each transformer also required a diode bridge to rectify the AC signal, duplicating those parts as well.
- The power supply is a linear one based on the LM7812 (Q1) which again is quite inefficient and not really suited to supply large currents to the lamps and bell.
- This board is a one-of-a-kind (sigh!) and making another is quite difficult (and expensive). If others would like to make them it is difficult to convey how to build it and to do so it would take some skill and patience. In fact nobody would probably want to do it.

Design Goals

With the above listed and numerous weaknesses in mind, the following design goals are set in order to address them;

- The hardware design should use commonly available (free) tools which make the design open and accessible to others who may want to build and or modify the design further.

- Building a copy of the circuit should be simple, enabling many people of all kinds of skill level to build one. With this in mind a Printed Circuit Board (PCB) will be produced using through hole components only.
- Overall part count to be reduced by 50%. Current design uses about 50 parts!
- Reduce overall cost to build. Current design costs $$$ to build.
- Trigger type and duration of signal operation shall be flexible (i.e. via software)
- Flash cycle time should be flexible (i.e. via software). However, as a baseline the previous design ran at 0.722 Hz with a 50% duty cycle (693 msec on/off) time so we will start with that.
- Bell Trigger time should be flexible (i.e. via software). Again since the previous design triggered the bell on each flash of the lamp, the bell will be triggered every 693 msec.
- Power stage should use more efficient power switching components (i.e. MOSFETs)
- Power supply shall be more efficient and provide enough power to lamps and bell (i.e. switching power supply)
- Software will control operation making it much more flexible to all users. Thus the controller will be based on a small and inexpensive microcontroller.
- Optional household AC or battery powered. This is due to the fact that unit may be used outdoors or where AC line current may not be available. In fact it is better to simply design the circuit to use DC power then add a AC->DC converter if needed.
- Learn something!
- Make it better!

Block Diagram

The block diagram below shows the overall control circuit as a black box. What is inside the black box is what needs to be re-designed.

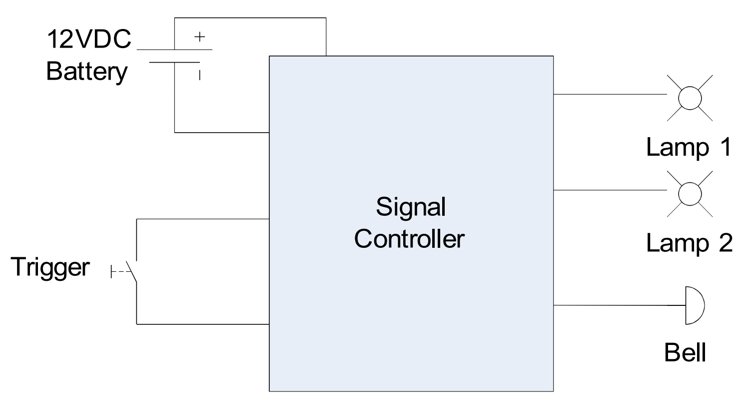


Illustration 1: Controller Block Diagram

Physical Requirements

To meet the design goals set forth earlier, some physical requirements must also be met;

1. Unit will be powered by a 12-VDC (nominal, 13.6 actual) battery
    a. Input power supply may vary between 9-VDC and 18-VDC and thus power supply

should be tolerant of that.

   b. Target battery is a commonly available motorcycle/tractor battery (need specs)
      - b.i. Consider small motorcycle battery with 3Ah capacity. High capacity are 14Ah capacity.
      - b.ii. Power Wheels batteries are typically 9.5Ah SLAs.
      - b.iii. 9V Duracell Copper Top, 10mA load over 50+ hours.
   c. Idle, quiescent current consumption should be less than 10mA.
2. Unit will include reverse polarity protection as to not blow out the controller or power supply if battery voltage is applied incorrectly.
3. Power supply will indicate operation using an LED.
4. Unit will be designed to anticipate being placed in a small enclosure (water resistant if used outdoors)
5. Must be able to independently control two automotive tail-light bulbs.
   a. May support either single or double filament bulbs
      - a.i. Tail/marker Filament 12-VDC @ 0.5 A
      - a.ii. Signal Filament 12-VDC @ 2.0A
6. Must be able to actuate a bell (Ace #36487)
   a. Note! *This bell is a self-interrupting circuit as purchased. A modification must be made to the bell to allow it to work as a bell in this project and not as a "door-bell"*. See Section 5 for more details.
   b. Bell is typically powered using three 6V DC batteries or AC-125C/36485 transformer (Output: 8 VAC-10 VA, 16 VAC-10 VA, 24 VAC-20 VA).
7. Must be triggered by a momentary switch.
8. Support a low-power mode when not in operation.
9. Diagnostic LEDs should be included for each lamp and bell to verify operation if such devices are not connected.
10. PCB and wiring harness should use commonly available wire-to-wire connectors suitable for outdoor environments.

Functional Requirements

The unit has very few functional requirements. Those requirements are detailed below.

1. **Trigger** – Actuates the grade crossing signal controller, causing lights to flash and bell to ring along with general amusement by all.
   a. Momentary actuation via push button or make/break connection.
   b. Toggles signal on/off
      - b.i. Press once to turn on, press again to turn off
         - b.i.1. Maximum on time is 30 seconds (timeout)
      - b.ii. Hold for continuous operation (> 1.5 seconds)
         - b.ii.1. No timeout (forever).
         - b.ii.2. Turn off if pressed again.
2. **Lamps** (two)
   a. 12-VDC Automotive Taillight Lamps (0.5/2.0 Amps)
   b. Flash at an approximate 2/3 second rate, 50% duty cycle.
3. **Bell**
   a. 12-VDC Door Bell (modified, 2-3 Amps peak pulse driven)

b. Strike bell in sync with each lamp flashing
c. Strikes are generated by short pulse to bell.

Trigger Design

The trigger is the only control input into the signal controller. It provides several functions although it is only a single momentary switch. To adequately define this behavior it is often helpful to develop a finite state-machine.

Modes: Active (Timeout, Continuous), Inactive

TODO: Include state-machine diagrams.

Stock Bell Modifications

The signal's bell to be used is a commonly available door-bell (Ace #36487) available at most hardware stores. The bell, as it comes stock in the package is actually made from an electromagnet which is used to operate a self-interrupting circuit. This is useful when operated from a line transformer (as is intended), but for our purposes the controller circuit will be actuating the bell instead through a short pulse. To do this, we will need to modify the bell slightly.

Here is an illustration of the door-bell as it is provided. Understanding its operation as a self-interrupting circuit is helpful so that we can understand how to modify it properly.

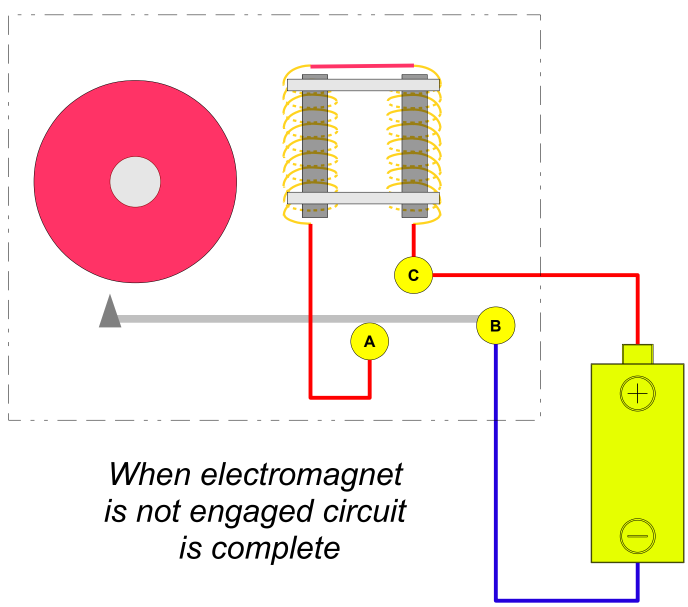


Illustration 2: Door bell (unmodified) when circuit is closed.

One end of the electromagnet wire is connected directly to one end of the electrical circuit (terminal-C). The other end of the wire connects to a metal contact, which is adjacent to a moving contact arm or clapper (terminal-A). The clapper is a thin piece of light, conductive metal. The anchored end of the clapper is wired to the battery (terminal-B). When the electromagnet is turned off (Illustration 2 above), the free end of the clapper rests against the contact point (terminal-A). This forms a connection between that end of the wire and the electrical circuit. In other words, electricity can flow through the electromagnet when the circuit is closed (terminal-A and B are connected).

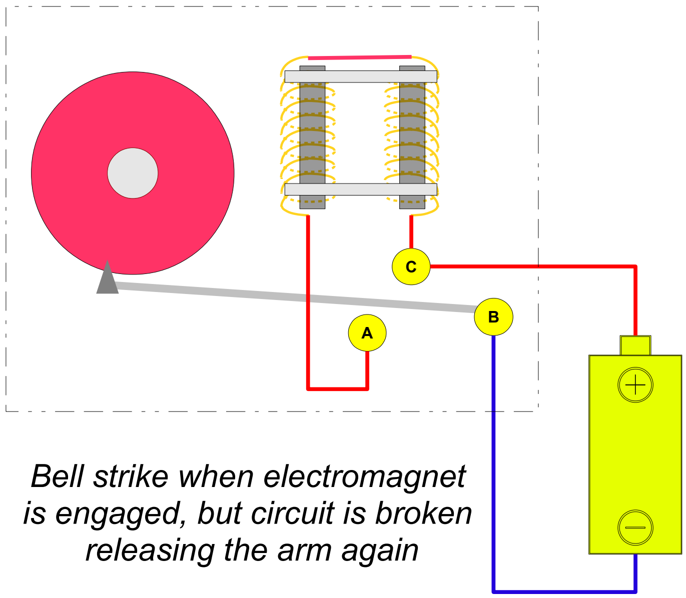


Illustration 3: Door bell (unmodified) when circuit is open

Initially, the electromagnetic field attracts the clapper off the stationary metal contact (terminal-A). This breaks the connection between the circuit and electromagnet, so the electromagnet shuts off (Illustration 3 above). Without a magnetic field pulling it back, the clapper snaps back into position against the stationary contact (terminal-A). This reestablishes the connection between the electromagnet and the circuit, and the current can flow through it again. The magnetic field draws the clapper up, and the process repeats itself as long as you apply power from the battery. In this way, the electromagnet keeps shutting itself on and off. The ringing sound you hear is the sound of the rapidly moving clapper hitting the bell dozens of times a second. This is the same system used in old-fashioned fire alarms and school bells.

This is all very interesting but it is not the behavior we want. What we want is to be able to control the bell enough to achieve a single strike of the bell each time a lamp is turned on. To do this we need to modify the stock door bell so that it is now a non-interrupting circuit. The single strike will instead be controlled by our controller.

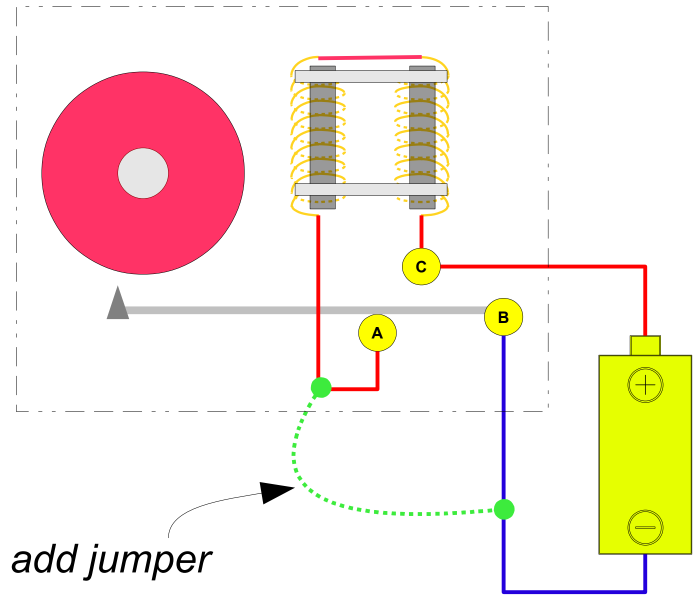


Illustration 4: Door bell (modified) as a non-self interrupting circuit

In Illustration 4 above we can see that by adding a jumper between terminal-A and terminal-B we effectively disable the self-interrupting part of the circuit. Now applying the power source to the circuit causes the clapper to strike the bell. Keep in mind with this modification, as long as power is applied, the electromagnet is engaged and the bell is struck. The clapper arm will not move away from the bell until the power source is removed. It is this operation that our controller is responsible for. The controller must be able to apply the power source to the circuit as a quick pulse, just enough to ring the bell once and then disengage. To determine just how long to pulse the circuit was unknown, so a test circuit needed to be designed.

Bell Driver Test Circuit

The bell driver portion of the design was always the most troublesome for me, so let's start there. Once we have that designed the rest should be a piece of cake.

The bell we are going to design the output circuit for is an Ace #36487. It consists of an electromagnet (coil) which must be stimulated with a voltage to make the arm move. The moving arm is attached to a long clapper, which rests alongside a circular bell. When the arm is moved (current applied) it strikes the bell. When current is removed the arm moves away from the bell (snaps back) ready to strike again.

The specs on the door-bell state that it can be driven with three 6V DC batteries (18 VDC) or AC using the 125C/36485 transformer. Since our controller uses DC voltage (supplied by a 12 VDC battery) we will use 12 VDC to drive the bell. It has been found that the bell can be adequately driven using 12 VDC.

Moving the arm with the electromagnet is problematic because it is what is known as an inductive load.

So some extra considerations need to be made when driving an inductive load versus and purely resistive one (such as a lamp). One thing to note is that the coil resistance at steady state is approximately 3-4-ohms. At steady state, with 12 VDC applied, this means the current through the coil would be as high as 4 amps. That also means the bell could draw as much as 36 watts! That is a significant amount of power. With power comes heat, and if you apply that current for too long the bell's coil will burn out.

Fortunately for us we actually don't want to drive the bell's coil with a steady current. In fact doing so would slam the arm against the bell and keep it there. There would not result a "free ringing" sound and instead it would be muffled by the arm resting against the bell. Instead we want a quick, bright ring when we trigger it. We can do this by driving a short pulse into the bell.

So how long should the pulse be? What will be the current through the bell during that time?
To answer this we need to experiment. A sample test circuit to determine this was constructed as in Illustration 5. This circuit allows me to send a pulse (from a microcontroller) into the bell driver circuit and view on an oscilloscope how much current is going through the bell. Current is measured across R_{ref} which is a 1Ω 10-Watt Power resistor.

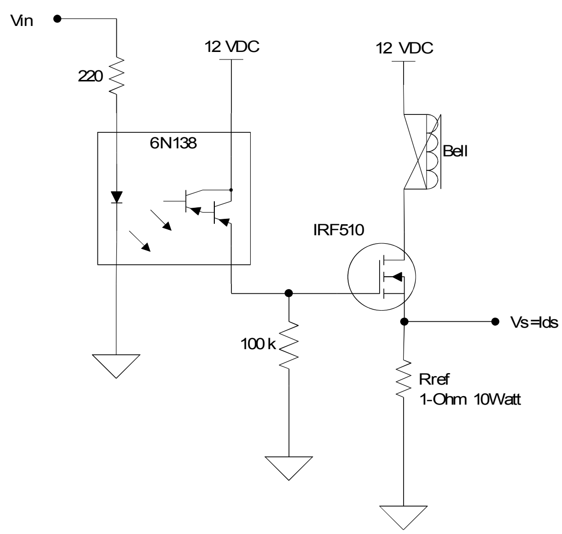


Illustration 5: Bell Coil Driver Test Circuit

To stimulate the bell, I drive a periodic (1-second) pulse of approximately 10 msec in width through the coil. Probing across R_{ref} shows the current through the 1Ω resistor. At this point the math is easy as we simply use V = I*R given R_{ref} = 1Ω, we then know the voltage is equal to the current through the resistor V=I.

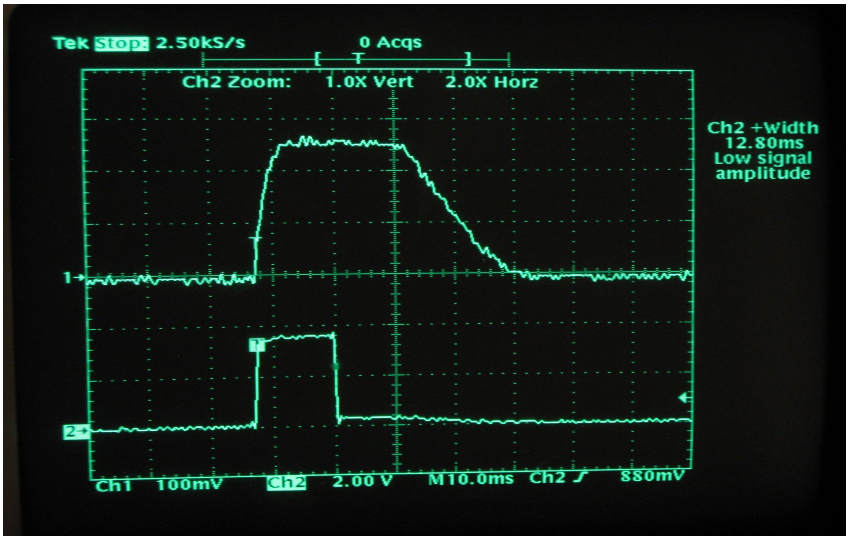


Illustration 6: Experimental Bell Driver Test Circuit Trace

The trace is shown above in Illustration 6 as seen on the oscilloscope. The microcontroller generates a 12.80 msec pulse to drive the opto-isolator which drives the MOSFET gate. The current through the coil is the same as that which is through the MOSFET drain-source and the R_{ref} resistor. As can be seen from the trace, The initial current starts at 0 and increases exponentially to 2.6-Amps, after about 4 msec reaches steady state, and then about 25 msec after the pulse goes low it begins to exponentially decrease back to 0 amps after about 43 msec total.

To simplify let's just assume with an approximate 10 msec pulse, the device sees full current for 50 msec every 700 msecs. Also recall that the 1Ω resistor, R_{ref} won't be in the actual drive circuit (it was only used for these measurements), so we have to assume the maximum current which we have seen at 12V, steady state which is about 2.75 A.

With this information we then know that at 12 VDC, the bell sees 2.75A for less than 10% of the time. This is worst case scenario and will serve as guidance into our design process for the bell driver.

Basic Design Elements

The overall circuit consists of 4 basic sections or elements which are 1) High Power Output Stage 2) The Controller 3) Trigger and 4) Power Supply. Each of these elements are addressed below.

High Power Output Stage

The high power output stage needs to drive two lamps and a bell. So we need three separately and digitally controlled high-power output control circuits. Since the microcontroller will be controlling and thus driving each circuit, the circuit must be controllable using a +5V signal. We also must consider power dissipation of the control circuit itself. We are switching (worst case) 2A of current to each lamp. Designs for the lamps and bell will be addressed separately below.

Often the first solution a design reaches for when needing to switch high power loads with a microcontroller or other small voltage and current device is a relay. However relays are notoriously unreliable in an outdoor environment, involve mechanical parts and will require drive circuitry of its own anyway since a microcontroller cannot on it's own drive the relay's coil.

Another approach is to use bipolar transistors. This is the approach I took on my first design using a TIP31 for the lamps and Darlington TIP120 for the bell. It also required a small NPN transistor to drive each of them. In total 16 parts (resistors and transistors) were needed to implement the circuit. Furthermore these transistors are current controlled devices requiring a significant base current to turn them on. Additionally the lowest attainable on-state voltage drop is determined by the collector-emitter saturation voltage $V_{CE(SAT)}$, which for the TIP31 $V_{CE(SAT)}$ = 1.2 V when I_B = 375mA, while the TIP120 $V_{CE(SAT)}$ = 2.0 V when I_B = 12mA. Power dissipation (which is converted to heat) is determined by P = VI so the heat generated by the part is subject to this $V_{CE(SAT)}$ parameter. Using the TIP31 to drive a 2A lamp results in a best case power dissipation of 2.4 watts (a 0.5 A load dissipates 0.6 watts). That best cast occurs when driving the base with 375 mA. A TO-220 package has a typical thermal resistances of $T_{R(j-c)}$ = 3 ºC/W and $T_{R(c-a)}$ = 62.5 ºC/W. Thermal resistance is an important parameter because it determines how high the temperature of the device will rise when dissipating an amount of power. Of specific importance here is the device's junction temperature ($T_{\theta j}$), which for the TIP31 cannot exceed 150 ºC. Plugging those values into the basic device thermal characteristic equations is as follows;

$$T_{R(j-a)} = T_{R(j-c)} + T_{R(c-a)} = (3 + 62.5)^\circ C/W = 65.5^\circ C/W$$

$$T_{rise} = T_{R(j-a)} \cdot P_D = 65.5^\circ C/W \cdot 2.4\,W = 157.2^\circ C$$

$$T_{rise} = T_{R(j-a)} \cdot P_D = 65.5^\circ C/W \cdot 0.6\,W = 39.3^\circ C$$

Woah! At an ambient temperature of 25 ºC the 2A lamp causes the TIP31 junction temperature to be about 182.2 ºC (360 ºF). That is 32.2 ºC above the maximum. The 0.5A lamp will make it about 64.3 ºC (148 ºF) which is within the limits. So, the 2A load is simply not possible (the transistor will blow) and the 0.5A load is still pretty hot. I am glad that this transistor is on only half the time! I have noticed that in the original prototype circuit these lamp drive transistors were pretty hot at the 0.5A load, I never tried a 2A load. Glad for that!

For the re-design I chose to use MOSFETs. MOSFETs are great for switching high currents because they are voltage controlled devices and if driven properly have a very low "on" resistance known as $R_{DS(ON)}$. Low "on" resistance means more efficiency, less power dissipation and therefore less heat. There are other choices as well (i.e. IGBTs for instance) but I decided on MOSFETs since much of this work would also be a basis for other work I would like to do in the area of motor controllers. So I went with MOSFETs.

There are some important considerations to take into account when driving MOSFETs however. One is the gate drive voltage, V_{GS}, the other is switching speed.

MOSFET based power designs can gain additional efficiency by properly matching the gate drive voltage V_{GS} with the MOSFETs being driven. Driving a MOSFET gate with a higher voltage, results in lower on-resistance, $R_{DS(ON)}$. However we cannot drive the MOSFETs with too high of a gate voltage without requiring additional drive (level translation) circuits which again add cost (the BOM) and complexity (the PCB layout). Higher gate drive voltage levels place additional charge into the gate-to-source junction of the MOSFET, resulting in increased losses within the MOSFET driver stage. To increase efficiency the applied voltage should drive the MOSFET gates such that the added gate charge and switching losses are less than the power savings gained by lowering the $R_{DS(ON)}$.

Higher gate drive results in higher switching losses when used at higher frequencies, switching losses dominate in which case a lower V_{GS} would be preferred. At lower frequencies, where conduction losses dominate, a lower $R_{DS(ON)}$ is preferred, and higher V_{GS} is desired. The consideration for switching speed is not really an issue in this application. The lamps are not driven faster than being turned on or off every 0.7 seconds (< 1.5 Hz). The bell is switched on and off a bit faster, but infrequently, once every 0.7 seconds it is switched on for 10 msec then switched off. Even if this were continuous it would only be at a rate of about 100 Hz. At those frequencies switching speed really is not significant. It usually isn't a factor with MOSFETs till we start talking about several kHz switching speeds.

In this application, the controller has a 5V output so ideally being able to drive a MOSFET at a V_{GS} of 5V is desired.

Another consideration is whether to use a low-side or high-side driver configuration.

A low side driver is implemented with the switch (N-Channel MOSFET) between the device being controlled and ground. A high side driver is implemented with the switch (P-Channel MOSFET) between the +V supply and the device. These two types are shown below.

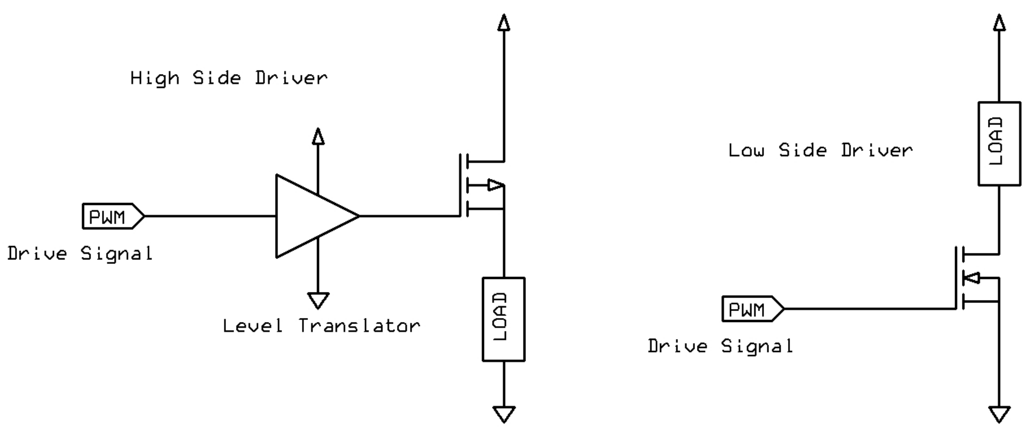


Illustration 7: Comparison of High-Side and Low-Side Drivers

Low-side drivers in some applications are considered somewhat “less safe” because the device has voltage on it even when turned off. For instance, what if someone cuts the cable or damages the device and it shorts to ground? Additionally since the load is electrically “hot”, it can be turned on without the controller by connecting the opposite side to ground. If this were a motor for instance, the motor could be accidentally turned on.

High-side drivers are preferred from a robustness and sometimes safety perspective. This is particularly important when the device being driven is not near the controller. When the high-side driver is off, the device is at ground potential, with little or no current flow. However, high-side drivers are somewhat more complex to implement and require more components (i.e. drivers, level translators), especially when the logic level driving them doesn't match the driver voltage.

In this case a low side driver was implemented for its simplicity and reduced complexity (shown below). The driver (without LED state indicator) consists of only two parts. The MOSFET, Q switch and bleeder resistor R_B. An LED diode, D and current limiting resistor R_A were added to given visual indication of the switch state which is helpful during development and troubleshooting. It also allows the circuit to be useful even without a load connected to the output.

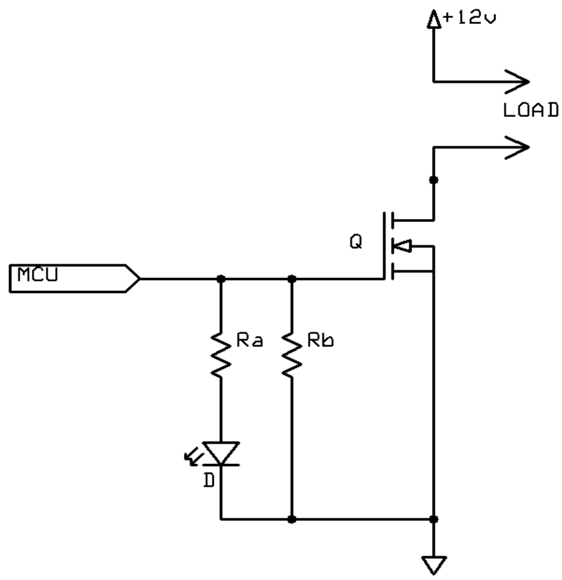


Illustration 8: Basic Power Output Low-Side Driver Circuit Design

The bleeder resistor, R_B is used to bleed off current from the MOSFET gate when not being driven. A MOSFET gate acts much like a small capacitor which can leave a charge (and thus leave the switch on) when not being driven. A value of 100kΩ was chose for this resistor which draws a maximum of 50µA. In fact the bleeder resistor is probably not even needed in this application since it is connected to a microcontroller output pin. A bleeder resistor is only really needed when the input maybe left floating.

Note: Rev.B boards do not have this bleeder resistor as it is not needed for this application.

The indicator circuit made up of a red LED, D (SSL-LX5093HD) and current limiting resistor R_A = 1kΩ sets the LED current at about 3 mA due to the LED's forward voltage drop of V_F = 2 V. See Section 'Power Supply Design' below. This is sufficient current to drive the LED.

MOSFET Q was selected as an On-Semiconductor [NTD4906N-1G] N-Channel Power MOSFET. The characteristics of this MOSFET are low $R_{DS(ON)}$, low gate capacitance and relatively small through-hole package (IPAK). At $0.26/ea they are very inexpensive.

Absolute Maximum Ratings at T_A = 25ºC

V_{DSS} = 30 V
V_{GS} = +/- 20 V
I_D = 10.3 A (continuous, no heat sink or pad)
P_D = 1.38 W
$T_{\Theta j}$ = 175 ºC

Electrical Steady State Characteristics

$R_{DS(ON)}$ = 7 mΩ at V_{GS} = 4.5 V
$R_{DS(ON)}$ = 6 mΩ at V_{GS} = 5.0 V
$T_{\Theta(j\text{-}a)}$ = 109 ºC/W

It would be perhaps considered "better-design" to have implemented a high-side driver instead. But its not like we are dealing with big heavy motors here. So I went with the simpler design approach.

Lamp Output Stage Design

Note: Most automotive trailer bulbs have two terminals each connected to a different filament. One for turn signal (green wire) and another for taillight (brown wire). Although I have not found any consistent or universal wire colors for this, you will need to investigate what they are based on the unit you have. They are both however referenced to the chassis ground or frame via a bolt. The turn signal draws approximately 2A while the taillight draws 0.5A. I have used both terminals and obviously one is significantly brighter than the other. Since this is likely to be used outdoors, the brighter one is considered in the design. Also since this would also represent the worst case scenario, the 2 A filament is used for that reason as well.

Using I_D = 2 A, and $T_{\Theta(j\text{-}a)}$ = 109 ºC/W we calculate power dissipation, junction temperature rise and expected junction temperature in ambient conditions T_A = 25 ºC.

$$P_D = I_D^2 \cdot R_{\mathrm{DS(ON)}} = (2\mathrm{A})^2 \cdot 0.007\,\Omega = 0.028\,W = 28\,mW$$

$$T_{rise} = T_{\Theta(j-a)} \cdot P_D = 109^\circ C/W \cdot 0.028\,W = 3.052^\circ C$$

$$T_{\Theta j} = T_a + T_{rise} = 25^\circ C + 3.052^\circ C = 28.052^\circ C$$

Both power dissipation, P_D = 28 mW and junction temperature $T_{\Theta j}$ = 28 ºC (83 ºF) are well below this MOSFETs maximum ratings, $P_{D(max)}$ = 1.38 W and $T_{\Theta j(max)}$ = 175 ºC. These results are also much better than the original transistor based design.

Bell Output Stage Design

Using I_D = 2.75 A, and $T_{\Theta(j\text{-}a)}$ = 109 ºC/W we calculate power dissipation, junction temperature rise and expected junction temperature in ambient conditions T_A = 25 ºC.

$$P_D = I_D^2 \cdot R_{\mathrm{DS(ON)}} = (2.75\,A)^2 \cdot 0.007\,\Omega = 0.05294\,W = 52.94\,mW$$

$$T_{rise} = T_{\Theta(j-a)} \cdot P_D = 109^\circ C/W \cdot 0.053\,W = 5.77^\circ C$$

$$T_{\Theta j} = T_a + T_{rise} = 25^\circ C + 5.77^\circ C = 30.77^\circ C$$

Both power dissipation, P_D = 53 mW and junction temperature $T_{\Theta j}$ = 30.77 ºC (87.39 ºF) are well below this MOSFETs maximum ratings, $P_{D(max)}$ = 1.38 W and $T_{\Theta j(max)}$ = 175 ºC.

This circuit is identical to the lamp output stage expect for the fact that a "freewheel, flyback, or snubber diode" is needed in a reverse biased direction across the bell's coil. This helps to eliminate the sudden voltage spike seen across an inductive load when its supply voltage is suddenly removed. When the device rapidly turns off, the diode becomes forward biased and the collapse of the energy stored in the coil causes a current to flow through the freewheel diode. Without this protection in place high currents would occur causing a high voltage spike or transient to flow around the circuit possibly damaging the switching device. This freewheeling diode conducts this voltage spike, effectively shorting the coil, protecting the MOSFET switch. The diode absorbs the energy (power) dissipating it out of the coil, for this reason the diode needs to have a high enough power dissipation capability to adequately absorb the power when the switch is turned off.

Generally you need a diode with a high surge current capability to provide this protection. A 1N4007 general purpose rectifier is used to provide this protection. The 1N4007 is very common diode (only $0.09 ea) and has specifications such as I_{FSM} = 30A, V_{RRM} = 1000V and V_F = 1.1V.

Note: A faster diode such as the popular 1N5819 may be desired although it has a much lower V_{RRM} of about 40V. However since we are interested in suppressing spikes and not a constant reverse voltage these may work fine. Schottky diodes are optimized for very low voltage drop, and fast operation. The 1N5819 features a maximum P_D = 1.25 W, I_{FSM} = 25A, and V_F of 0.6V. For this reason these 1N5819 diodes are used in the Rev.B design instead.

The Controller

The functional requirements of this project are very simple and thus some might argue a microcontroller is overkill. That may be true, but simple microcontrollers these days are very inexpensive, easy to use and much more flexible than their discrete hardware equivalent implementation (i.e. 555 timer, oscillating comparators, RC-circuits, etc). Also writing a few lines in software which could in turn be easily modified, enhanced or completely re-purposed might be beneficial to someone. For these reasons, a microcontroller was chosen opposed to discrete hardware to implement the controller logic.

There are countless controllers on the market and many of which are very low-cost, low-power and have more than enough RAM, ROM and I/O resources to meet this very simple application's needs. However we need to start somewhere. To start we need to derive the basic controller requirements (define the needed resources). These resource requirements are derived from Section - Functional Requirements and are listed below;

- 5V Supply
- 1 Input (switch)
- 3 Outputs (bell-1, lamps-2)
- 1 Time Base (for timers)
- ROM (< 256 bytes) - estimated
- RAM (< 64 bytes) - estimated

Hundreds of microcontrollers meet these requirements. The first step in narrowing down the choices might be to narrow down a vendor. I have familiarity working with Freescale, TI and Microchip controllers to name only a few. Each of those vendors offer low end controllers more than capable of meeting this application's resource requirements. The Freescale stuff I have worked with in the past has been in the HC908-QT/QY family which is largely outdated at this point, not a good choice for new designs.

TI also has an insanely inexpensive and convenient TI MSP430 LaunchPad. The TI LaunchPad is an easy-to-use development tool intended for beginners and experienced users alike for creating microcontroller-based applications. At $4.30, the LaunchPad offers everything you need to get started with your projects. Indeed it does! You get a development board with LEDs, Pushbuttons, a microcontroller(MSP430G2231) and debugger. How can you go wrong? Plus this project will then provide a simple introduction to using this controller and platform, which would meet my goals of actually learning something new. However the down side is this platform runs on 3.3V supply. Although it could be made to work, we really need 5V going to our MOSFETs (see above section). So something else is needed.

Enter Microchip, if for no other reason, I have this handy PicKit3 debugger/programmer and demo board laying here and I really have not had the opportunity to try it out. Plus I really have not given Microchip components much of a try either, so it would be interesting to see what they offer. Microchip also offers a free set of development tools, the MPLAB X IDE which will suit me fine. Again, something free and new to learn...all systems go.

So what do they have? A parametric search of what they have available via their product selector yields 25 parts which are 8-bit architecture and up to 8-pins all in their PIC12F family of parts. Off to Mouser and of the PIC12F family of parts with a PDIP-8 package (because we want it easy to solder and prototype) we find those same 25 parts. So which is the cheapest? At $0.79 for a single unit, the PIC12F508-I/P is the winner.

The PIC12F508 features the following:

- 4-MHz operation with internal precision oscillator
- In-Circuit Serial Programming (ICSP) and In-Circuit Debugging (ICD) Support
- Power-On Reset (POR)

- Internal Weak Pull-Ups on I/O Pins
- Wake-up from Sleep on Pin Change
- Wide Operating Voltage Range: 2.0V to 5.5V
- 6 I/O Pins:
    - 5 I/O pins with individual direction control
    - 1 input only pin
    - High current sink/source for direct LED drive
    - Wake-on-change
    - Weak pull-ups
- 8-Bit Real-Time Clock/Counter (TMR0) with 8-Bit Programmable Prescaler
- 512 Words of FLASH
- 25 Bytes of SRAM

This should more than meet the resource requirements of the controller. The other nice thing is that if at some point in the future I decide I need more RAM and/or FLASH all of the parts in this family are pin-compatible. Specifically the PIC12F1840-I/P at $1.41 would be a suitable drop-in replacement at the higher end of the memory resources available with 256-bytes of RAM and 7K of FLASH in case my original estimates are wrong.

The pin-out of the PIC12F508 is as shown below. Minimal connections are required as nothing special is required beyond a stable power supply (5V) connected between V_{DD} and V_{SS}.

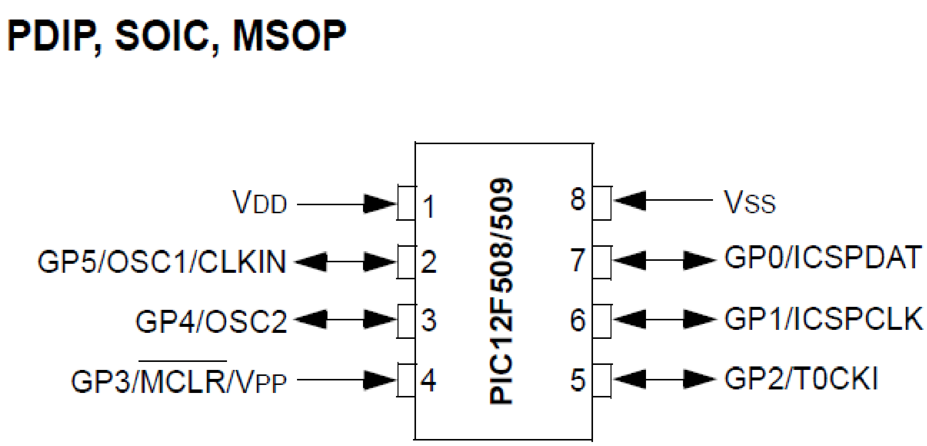


Illustration 9: PIC12F508 Pin out Diagram

Pin assignments are assigned as follows (TRIS Register):

MCU Pin	Pin #	Function	Feature
GP0/ICSPDAT	7	Output	Unused
GP1/ICSPCLK	6	Output	Unused
GP2/T0CKI	5	Output	Bell
GP3/MCLR/VPP	4	Input	Trigger
GP4/OSC2	3	Output	Lamp-1
GP5/OSC1/CLKIN	2	Output	Lamp-2

Note this leaves us with 2-unused pins which may be re-purposed later. For now they will be designated as outputs driven low (unused). Note PIC® Microcontroller Low Power Tips 'n Tricks **TIP #3 Configuring Port Pins** states "*If a port pin is unused, it may be left unconnected but configured as an output pin driving to either state (high or low)...*" and "*There is no additional current consumed by a digital output pin other than the current going through the pin to power the external circuit.*". Therefore leaving the pin unconnected should not consume any additional current.

The input trigger was assigned to GP3 because GP3 can be used as an input only. Since the functionality of this pin is limited, it makes sense to assign it to the trigger. GP3 can also be configured for "wake-up on change" which can be useful when implementing low-power modes of the MCU.

Device DC Electrical Characteristics

Voltage on VDD with respect to VSS ....0 to +6.5V
Voltage on MCLR with respect to VSS......................................0 to +13.5V
Voltage on all other pins with respect to VSS .... -0.3V to (VDD + 0.3V)
Total power dissipation ....800 mW
Max. current out of VSS pin ....200 mA
Max. current into VDD pin ....150 mA
Input clamp current, IIK (VI < 0 or VI > VDD)......................................+/-20 mA
Output clamp current, IOK (VO < 0 or VO > VDD) ....+/- 20 mA
Max. output current sunk by any I/O pin ....25 mA
Max. output current sourced by any I/O pin ....25 mA
Max. output current sourced by I/O port ....75 mA
Max. output current sunk by I/O port ....75 mA

DC Characteristics of the Inputs and Outputs

The PIC12F508 will recognize a high state when the input is between 2.0V → VDD. It will recognize a low state when the input is between VSS → 0.8V.

When configured as an output a pin set to high the voltage observed will be a minimum of VDD – 0.7V. Set to low state the output voltage observed will be a maximum of 0.6V.

PicKit3 ICSP and ICD support

As the datasheet indicates the PIC12F508 supports both In-Circuit Serial Programming (ICSP) important for programming the chip with software, and ICD (In-Circuit Debugging) important for development and debugging. However to get the ICD to function an additional “header” is needed. I don't know much about that “header” so since this software is very basic, I am hoping I will not need it.

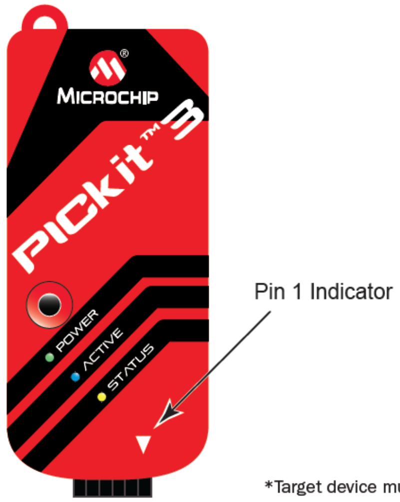
ICSP however is important, at least on a prototype or breadboard. This allows new software to be programmed into the MCU. It is not needed for normal operation. However it is convenient (and common practice) to at least layout the PCB with pins that can be populated with a header so that a ICSP can be inserted and program the device using the PCB rather than relying on some external programming board. For this reason the ICSP pins will be on the PCB but may be unpopulated if this capability is not needed or desired (cost).

Microchip provides a very handy “poster” which simplifies understanding of how to connect the ICSP pins to the microcontroller. It can be found here.

To summarize the connections on this PCB will be connected to the 5 pins of the PicKit3 via a 6-pin 0.1” pitch header as follows:

MCU Pin	MCU Pin #	PicKit3 Pin#	PicKit3 Function
V_{SS}	8	3	Vss Ground
GP0/ICSPDAT	7	4	PGD (ICSPDAT)
GP1/ICSPCLK	6	5	PGC (ICSPCLK)
GP2/T0CKI	5		
GP3/MCLR/VPP	4	1	$\overline{\text{MCLR}}/V_{PP}$
GP4/OSC2	3		
GP5/OSC1/CLKIN	2		
V_{DD}	1	2	V_{DD} Target

Note: PicKit3 Pin #6 (PGM(LVP)) is not need and thus not connected.

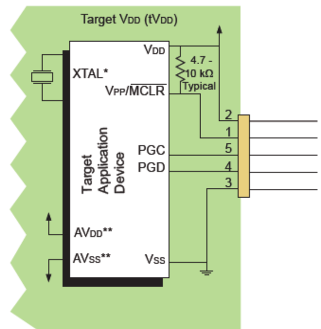


Target Connector Pinout

Pin	Signal
1	MCLR/VPP
2	VDD Target
3	VSS Ground
4	PGD (ICSPDAT)
5	PGC (ICSPCLK)
6	Do not connect*

*Reserved for future use.

Correct



Target Application PC Board

*Target device must be running with an oscillator for the debugger to function as a debugger.
**If the device has AVDD and AVSS lines, they must be connected for the debugger to operate.

Microchip offers a handy document entitled In-Circuit Serial Programming (ICSP) Guide which details how designers should incorporate this capability in their designs. The section *How to Implement ICSP Using PIC12C5XX OTP MCUs* on page 13 (TB017) is particularly useful which outlines the following;

During the initial design phase of the application circuit, certain considerations have to be taken into account (snap! if I only read that earlier). In order to implement ICSP on your application board you have to put the following issues into consideration:

1. Isolation of the GP3/MCLR/VPP pin from the rest of the circuit.
2. Isolation of pins GP1 and GP0 from the rest of the circuit.
3. Capacitance on each of the VDD, GP3/MCLR/VPP, GP1, and GP0 pins.
4. Interface to the programmer.
5. Minimum and maximum operating voltage for VDD.

Specifically points 1 and 2 above regarding isolation are particularly important. In this design GP3 is already sufficiently isolated since when the switch (trigger) is not pressed the output transistor is high impedance to ground and the only connection to GP3 is a pull-up resistor R2. GP1 and GP0 however should have isolation points which allow them to be driven by the programmer only when the ICSP mode is used.

Note: Unfortunately in the first design I did not consider this. In the Rev.B board I did separate the ICSP portion of the design so that it is more easily isolated and separated from the main circuit design. The ICSP implementation in Rev.A does work as long as the MOSFETs are not connected (particularly Q2). The Rev.B boards do not have this limitation.

MCU Configuration

The MCU Configuration shall enable the following via the __CONFIG() keyword;

OSC_IntRC – Use internal RC oscillator
WDT_OFF – No watchdog timer, disabled
CP_OFF – Code Protection is off
MCLRE_OFF – Don't use master clear reset function on GP3, disabled

The MCU OPTION Register value will be a value of 0xD7 described as follows;

```
OPTION = 0b11010111;
         //1------- Wake-up on pinchange disabled. (GPWU=1)
         //-1------ Weak pullups disabled. (GPPU=1)
         //--0----- T0CS, use internal instruction cycle clock Fosc/4
         //---1---- T0SE, don't care.
         //----0--- Prescaler assigned to TMR0 (PSA=0)
         //-----111 Prescaler 1:256 (PS=111) 256 usec
```

Note: Initially I didn't bother using the low-power mode of the MCU or enable the use of "wake-up on change". That is something which can be experimented with and added later.

Timer Configurations and Calculations

The internal precision oscillator will be used to reduce overall part count and cost. The internal oscillator operates as 4-MHz (F_{OSC}). There are 4-cycles required per clock tick so the rate at which instructions run is 1-MHz ($F_{OSC}/4$). Since the 1:256 timer prescaler is configured (see above) this means our timer will increment once every 256 usec, known as a clock tick.

$$t_{clock} = \frac{Prescaler \cdot 1000000\,usec}{\frac{F_{OSC}}{4}} = 256\,usec$$

For convenience our timer 'tick' should occur once every 10 milliseconds. So to get milliseconds from t_{clock}, we calculate t_{count} as follows.

$$t_{count} = \frac{10000\,usec}{t_{clock}} = 39$$

Back calculating the actual tick our application will experience using this timer we find that using following:

$$t_{tick} = t_{count} \cdot t_{clock} = 9984\,usec = 9.984\,msec$$

This ends up being about a -0.16% error which is pretty good. But since we now know what t_{tick} is, we can use that to compute things like;

$lamp_{timeout}$ = 0.7 seconds, $lamp_{count}$ = 70, error = -0.16%

$$lamp_{count} = floor\left[\frac{0.7\,sec \cdot 1000000\,usec/sec}{t_{tick}}\right] = 70$$

$active_{timeout}$ = 30 seconds, $active_{count}$ = 3004, error = -0.0269%

$$active_{count} = floor\left[\frac{30\,sec \cdot 1000000\,usec/sec}{t_{tick}}\right] = 3004$$

$lpress_{timeout}$ = 2 seconds, $lpress_{count}$ = 200, error = -0.16%

$$lpress_{count} = floor\left[\frac{2\,sec \cdot 1000000\,usec/sec}{t_{tick}}\right] = 200$$

The timer is RESET by writing a 0 to the TMR0 register and expires when the TMR0 register is read and is equal to t_{count} (39). Thus the basic "super-loop" pseudo code is as follows;

```
#define TIMER_EXPIRED (TMR0 == TMR0_COUNT)
#define TIMER_RESET (TMR0 = 0)

START

     FOREVER
     {
        // Wait for timeout (~10 msec)
        while (0 == TIMER_EXPIRED);
        TIMER_RESET;

        // Do some work...
     }

END
```

The above "super-loop" runs every 10 msec.

Trigger

To trigger the signal an SPST switch (or other suitable contact) is used to indicate the signal is to turn on. There are many ways to interface external inputs into a microcontroller. I have written a bit of a survey of various input protection techniques and concepts in my post; Microcontroller Input Protection Techniques which can be referenced for more detailed information on that subject.

Based on how this controller is intended to be used (outdoors, uncertain environments) I decided it would be best to use optical isolation as an input interfacing and protection technique. This technique obviously uses an optoisolator and that implementation is described in the following section.

Below is a simple implementation of an optoisolated switch input. Notice the output of the optoisolator is configured in “pull-up” mode, meaning when the output transistor is off, the output is high (V_{DD} = 5V) and the MCU input sees this as a logical “1”. When the output transistor is on, the output is low (ground) and the MCU sees this as a logical “0”. To turn the transistor on, the input LED must be driven by the switch input which happens when the switch is closed. The LED's cathode lead is driven to ground, completing the circuit from V_{DD} (5V) through R_F to ground. From the MCU software perspective, a press is indicated by a logical “0” and not pressed by a logical “1”.

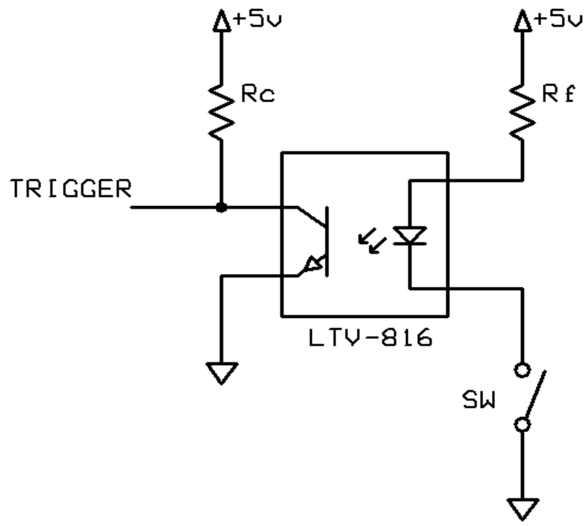


Illustration 10: Optically Isolated Input

The nature of the pushbutton trigger switch lends itself very well to be interfaced to the MCU through an optoisolator. The speed of the optoisolator is not critical since the pushbutton cannot be pressed very fast anyway. Therefore virtually any single channel optoisolator will do. Due to cost ($0.18/ea) and availability I chose the Lite-On LTV-816.

The relevant specifications of this part are as follows:

Absolute Maximum Rating (T_A = 25ºC)

Input: Forward Current (I_F) = 50ma max
Input: Reverse Voltage (V_R) = 6V
Output: Collector Current (I_C) = 50mA max
Output: Collector-Emitter Voltage (V_{CE}) = 80V max
Isolation Voltage: 5 kV_{RMS} max

The isolation voltage means that it can effectively isolate up to 5 kV_{RMS} on the optoisolators input into the MCU (sufficient for a nearby lightning strike, electrostatic discharge (ESD), etc.). Also the parameter V_R indicates the input can tolerate a reverse voltage of maximum 6 V, so if someone applies something other than ground to the input it can tolerate up to 11V (11V – 5V = 6 V). In which case only the optoisolator may blow, the MCU will still be protected.

Electrical/Optical Characteristics

Input Forward Voltage (V_F) = 1.2V (@ I_F = 20ma)

We would like the output of the optoisolator's transistor to be well in saturation mode with minimal forward current (I_F). We also need to make sure the transistor's output can provide enough current to drive the MCU input. Typically MCU inputs are very high impedance and require little to no current at all. I could not find any specification on minimal input current for our MCUs GPIO, but I assume it is OK if we meet the same current requirements that their own internal weak pull-up resistors offer. So we ill go with the maximum value of I_{PUR} of 400µA. I am sure the MCU needs much less. So as long was we can provide 400µA we should be fine. "driving" the MCU really isn't so much the issue as just simply making sure the optoisolator is sufficiently driven by the input LED to turn it on and off.

Given we want an I_C > 0.4mA. Figure 3 of the LTV-816 datasheet indicates that if I_F is greater than 2mA, V_{CE} is about 0.5V if I_C is < 0.5mA.

So for I_F > 2mA, V_F = 1.2V, using ohms law:

$$R_F = \frac{V_{\mathrm{IN}} - V_F}{I_F} = \frac{5.0\mathrm{V} - 1.2\mathrm{V}}{.002\,A} = \frac{3.8\mathrm{V}}{.002}\,A = 1900\ ohms$$

So as long was we we have no more than a 1900Ω resistor in series with the optoisolator LED, we will have achieved output transistor saturation. However let's just use a nominal R_F = 470Ω. We the back calculate to find I_F;

$$I_F = \frac{V_{\mathrm{IN}} - V_F}{R_F} = \frac{5.0\mathrm{V} - 1.2\mathrm{V}}{470\,ohm} = \frac{3.8\mathrm{V}}{470}\,ohm = 8.09\,mA$$

which again by Figure 3 in the LTV-816 datasheet is more than enough current to saturate the output transistor. If I_F = 8mA, $V_{CE(sat)}$ = 0.5V when I_C = 0.5mA

Note: Perhaps a 1-kΩ resistor should be used instead. Something to consider. If anything to just keep the different kinds of parts used to a minimum and would still meet the requirements.

If we use a nominal R_F = 1-kΩ. We the back calculate to find I_F;

$$I_F = \frac{V_{\mathrm{IN}} - V_F}{R_F} = \frac{5.0\mathrm{V} - 1.2\mathrm{V}}{1000\,ohm} = \frac{3.8\mathrm{V}}{1000}\,ohm = 3.8\,mA$$

which is still above the required forward current of 2mA. Thus a 1-kΩ resistor is used in the Rev.B design.

So now onto the output stage. Again using ohm's law we see that;

$$R_C = \frac{V_{\mathrm{IN}} - V_{\mathrm{CE(sat)}}}{I_C} = \frac{5.0\mathrm{V} - 0.5\mathrm{V}}{0.5\,mA} = 9000\,ohms$$

Again, using a typical value of R_C = 10kΩ, we see that;

$$I_C = \frac{V_{\mathrm{IN}} - V_{\mathrm{CE(sat)}}}{R_C} = \frac{5.0\mathrm{V} - 0.5\mathrm{V}}{10000\,ohms} = 0.450\,mA$$

This means with a value of R_F = 1kΩ and R_C = 10kΩ. We are able to sufficiently drive the optoisolator in the saturation region and deliver enough signal to the MCU input to trigger it. With the configuration shown in Illustration 10, the input is a simple current limiting resistor (R_F) and switch in series with an LED. In this configuration all the switch needs to do is ground the cathode of the LED inside the optoisolator and the LED will turn on. When the LED is turned on, the output transistor will be in saturation and switched on. When the transistor is switched on, this creates a path to ground and will signal the MCU input (signal going low) that the trigger(button) has been pressed. Releasing the switch will cause the LED to turn off (as well the output transistor) causing the input of the MCU to read high (V_{DD} = 5.0V) though the effect of pull-up resistor R_C.

Power Supply Design

The power supply exists for the sole purpose of supplying a regulated voltage level to the microcontroller only. This design does not use a regulated voltage for power output devices such as the lamps or signal bell. Instead raw input voltage is switched directly via the high power output stage to those devices. Thus the power supply design needs to consider the power requirements of the microcontroller and indicators exclusively.

The microcontroller needs a stable 5.0V supply, but can tolerate a supply from 2.0V up to 5.5V (absolute maximum rating is 6.5V).

Considering operation of the unit at 25 ºC, from Figure 11-1 of the microcontroller datasheet we see that at F_{OSC} = 4MHz and V_{DD} = 5.0V, I_{DD} = 650 µA to 1100 µA. In sleep mode, the microcontroller will consume between 0.34 and 0.37 µA at V_{DD} = 5.0V.

Table 10.1 of the datasheet indicates that the Supply Current (I_{DD}) is typically 0.625 mA (1.1 mA maximum) when F_{OSC} = 4MHz and V_{DD} = 5.0V. Power-down current (sleep mode) is 0.35 µA (2.4 µA maximum) at V_{DD} = 5.0V. This looks consistent with the data from Figure 11-1. So assuming worst case for the microcontroller, I_{DD} shall be 2 mA maximum.

However we also have some LEDs we need to drive. Specifically the power indication LED (green) and up to 3 signal indication LEDs (red). Worst case is again that all LEDs are driven at the same time, although this never happens during normal operation.

For the LEDs we have the following specifications from their datasheets;

$$I_F = \frac{V_{IN} - V_F}{R_F}$$, where $V_{IN} = 5.0V$

LED[1]	Color	V_F(V)	R_F(Ω)	I_F(mA)
SSL-LX5093GD	Green	2.2	1k	2.8
SSL-LX5093HD	Red	2	1k	3

With one green and three red LEDs the maximum current consumption is about 11.8 mA for all LEDs. Add this to the microcontroller 2 mA required we arrive at 13.8 mA. No sweat, designing for a reasonable yet well over engineered rating of 100 mA will give us plenty of power supply capacity, especially if we decide to modify or enhance the circuit later.

So the power supply we need to design is a 5.0 V, 100 mA, 1/2 watt power supply. The input voltage is expected to come from a small motorcycle or lawn tractor battery (nominally 12 VDC) but input voltage from a battery can very depending on charge level. Alternatively AC-line maybe used given a step-down transformer and bridge-rectifier is in place (See my Full Wave Rectifier Design).

A very popular switching voltage converter is the LM2574. The LM2574N-5.0G is a fixed 5.0 V version which can provide up to 500 mA with an input voltage from 4.75 V to 40 V which costs about $1.62 in single units. This should work perfectly for our application. The datasheet contains an excellent guide to help design the power supply and is illustrated below.

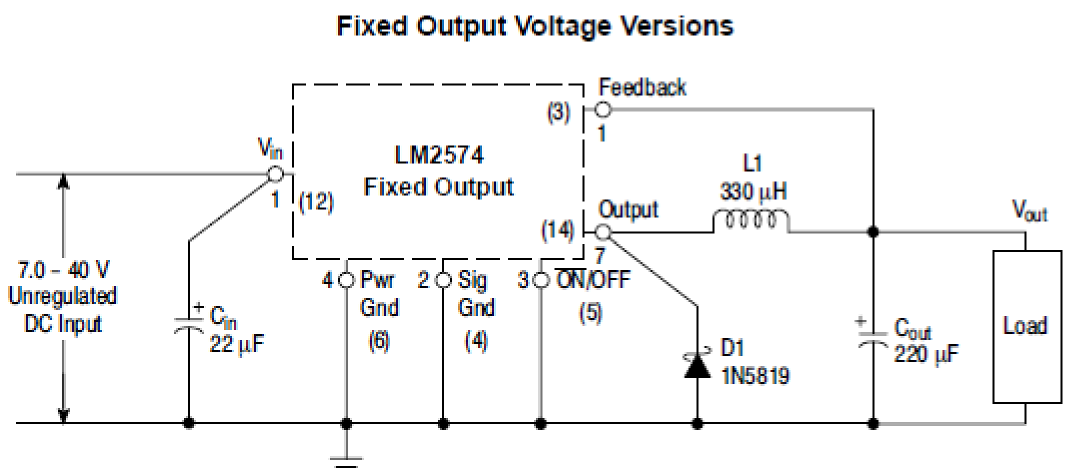


Illustration 11: Voltage Regulator Reference Circuit

Using the design guide (pg. 10) from the datasheet and given;

V_{OUT} = 5.0 V
$V_{IN(max)}$ = 18 V (typically 13.6 VDC from battery)
I_{LOAD} = 100 mA (over 7x fudge)

1 The Rev.B board replaced the 5mm LEDs (T-1 ¾) with a 3mm LED (T-1) that possesses the same operating characteristics.

We then need to find the values for C_{IN}, D_1, L_1, and C_{OUT} to complete our design.

C_{IN} - To prevent large voltage transients from appearing at the input and for stable operation of the converter, an aluminum or tantalum electrolytic bypass capacitor is needed between the input pin $+V_{in}$ and ground pin Gnd. This capacitor should be located close to the IC using short leads. This capacitor should have a low ESR (Equivalent Series Resistance) value. A 22 µF, 25 V aluminum electrolytic capacitor located near to the input and ground pins provides sufficient bypassing. I have chosen a Panasonic EEU-FR1H220, which is 22 µF, has a low ESR and rated voltage range of 6.3 VDC to 50 VDC.

D_1 = Fast recovery catch diode must be rated at > 1.2 x I_{LOAD} = 120 mA, but the datasheet still recommends to use the maximum the IC can handle (500 mA) thus 1.2x that is 600 mA.
Reverse voltage needs to be 1.25x maximum input voltage (22.5V). Therefore we need a 600 mA, 22.5 PIV catch diode. A 1N5817 low-drop power Schottky rectifier easily meets and exceeds this spec. The 1N5817 is very common diode (only $0.12 ea) and has specifications such as I_F = 1A, V_{RRM} = 20V and V_F = 0.45V. Note the V_{RRM} is very close to 22.5 PIV, but I think we are OK. If in doubt perhaps a 1N5819 could be used instead as it has a V_{RRM} of 30V. The 1N5819 has a V_{RRM} of 40V.

Note: Rev.B boards use a 1N5819 diode instead of the 1N5817. This will provide a more than adequate V_{RRM} of 40V and is nearly the same cost and exactly the same package as the 1N5817.

L_1 = Inductor Selection: Use the inductor selection guide shown in Figure 20 of the datasheet. From the selection guide, the inductance area intersected between the 10 and 20 V lines and 0.1A is mostly represented by 680 indicating a 680µH inductor is required. Current requirements are called out by the equation (at 500 mA). A suitable inductor is the Murata 12LRS684C 680µH, 0.50A maximum DC current inductor.

Note: I used 330µH, 0.63 A part only because it is a very common value and works well over a large output range. Prototype has shown that this value works fine in this application. However I replaced this with a 680µH 0.125A inductor in Rev.B to more closely match the actual power supply needs.

C_{OUT} = Output Capacitor: For stable operation and an acceptable ripple voltage, (approximately 1% of the output voltage) a value between 100µF and 470µF is recommended. Due to the fact that the higher voltage electrolytic capacitors generally have lower ESR (Equivalent Series Resistance) numbers, the output capacitor's voltage rating should be at least 1.5 times greater than the output voltage. For a 5.0V regulator, a rating at least 8.0V is appropriate, and a 10V or 16V rating is recommended. Therefore the part selected is a Panasonic EEU-FR1H221 which is 220 µF, has a low ESR and rated voltage range of 6.3 VDC to 50 VDC.

Note: A lower voltage rated capacitor could be used. A 16V version would be sufficient. Therefore Rev.B includes a lower voltage capacitor which is also a smaller package.

Finally with the power supply designed, we also consider input voltage reverse polarity protection. This feature was added because since this is to be used in the field, conditions may lend themselves to hot swapping a battery mistakenly in the reverse direction. Adding this protection is a simple effort. Battery input reverse voltage can be protected against by inserting a 1N5817 Schottky Diode in series with the power supply input. Even at worst case expected peak current draw of 0.1A, V_F=0.45V thus $P_D = I_F \times V_F$ = 0.045 watts. Rth(ja) = 100ºC/W, thus Tj = 4.5ºC + 25ºC(ambient) = 29.5ºC (the part can handle 150ºC max). We will loose about 0.45V off of our input power supply (and dissipate 0.045 watts of power, worst case), but we are providing at least a 12VDC supply anyway. So this should not be an issue.

Note: The Rev.B board uses a 1N5819 to keep the number of different parts required to a minimum and still meet (and actually exceed) the input reverse voltage protection needs. The 1N5819 does have a slightly higher forward-voltage drop, V_F = 0.6V (resulting in an extra 0.015 watts) but we still have more than enough headroom from a 12V battery supply.

The required output of this power supply is very low (< 20 mA) and therefore power dissipation is very small. For this reason heat sinking was not considered in this design.

PCB Design

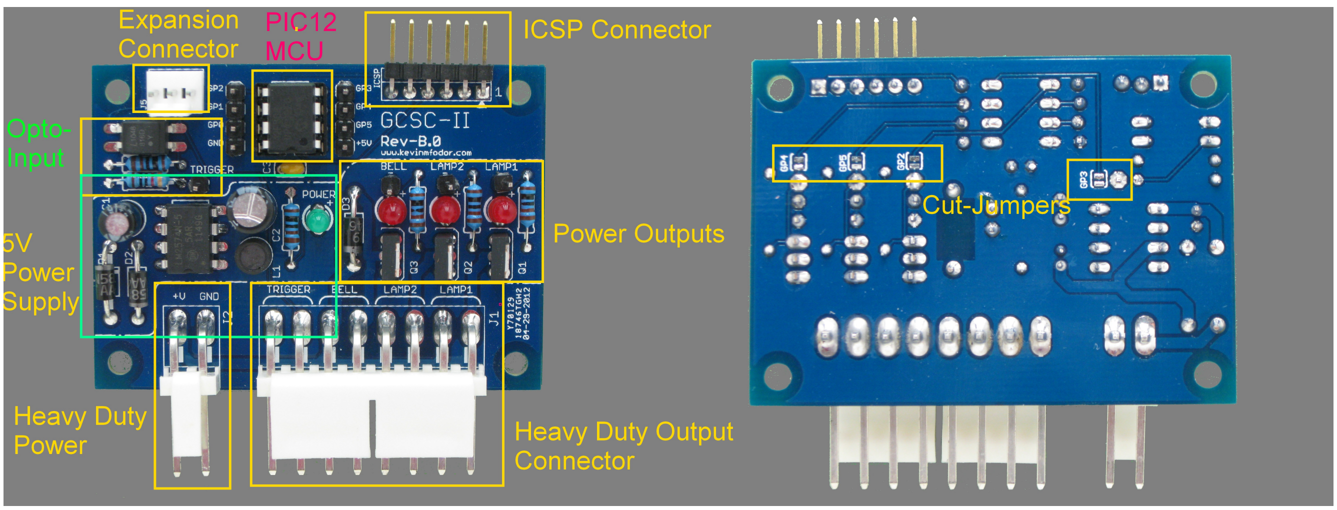


Illustration 12: Rev.B PCB Assembled with Annotations

For this design Cadsoft's EAGLE PCB design software will be used. This was chosen for its popularity and availability for hobbyist, non-commercial designs. It offers convenient schematic capture with integrated PCB layout tools.

As for physical size of the PCB, it would be nice for it to fit inside the 1-1/2” SCH-40 PVC Pipe. The inside diameter is a nominal value 1.610”. However if not possible this OK as when an external power supply is required it is likely to be mounted in an outside water-resistant enclosure anyway. Due to increased connector sizes this goal was not achievable on the Rev.B boards. This is OK since it turned out to be much more convenient to mount the board inside an outdoor electrical box which easily mounts onto the signal mast as shown here.

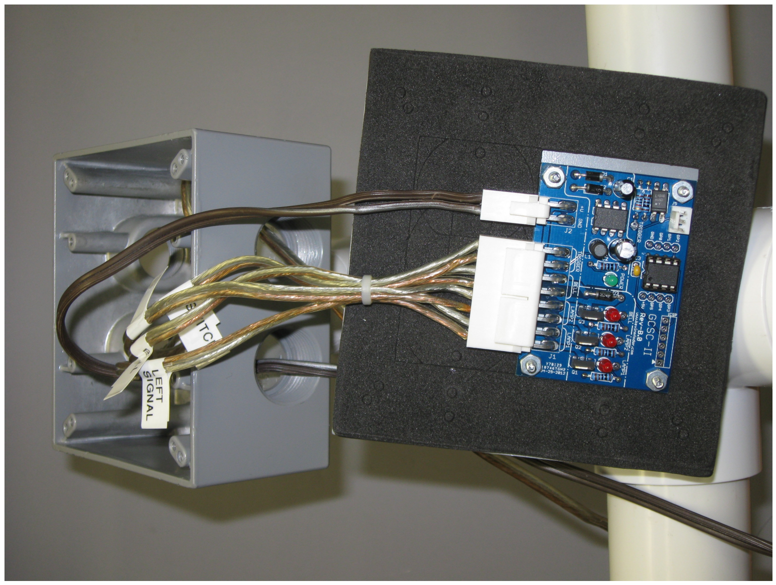


Illustration 13: Rev.B PCB Shown Mounted on the Mast in an Electrical Box

The PCB shall provide holes to accommodate four #4 Phillips button head screws for mounting inside an enclosure or other rigid surface. PCB material is 2-Sided, FR-4, 1-oz/ft^2 (35µm) copper board thickness of 1.6mm. These are the most common specifications of 2-sided PCB board house manufacturers.

All components will be through-hole parts to facilitate simple assembly and moderate soldering skills.

Working grid is 50 mils, with 10 mils routing to grid. Basic single tracks (width/spacing, mils) are;

- 10/10 signals
- 20/20 for V_{CC} and GND,
- 40/40 for Battery (assumes ~2.4 Amps)

Trace minimum width guidelines for 1-oz/ft^2 (35µm) copper, 10 ºC rise;

2A, T_{width} = 31.3mil
3A, T_{width} = 55.0 mil
4A, T_{width} = 82.7 mil

Ground and Power Plan Pour is 0.012 mil width, 0.012 mil isolate with thermals and orphans.

Larger tracks will be used wherever possible.

Software Design

Like most control systems, their implementation lends itself well to a finite state machine (FSM). If implemented properly they are easy to understand and maintain.

TODO: Main-loop, Input Debounce, Signal FSM

Retrospect

In other words; things I thought about between Rev.A and Rev.B

In the process of developing, building and testing the Rev.A version of the GCSC-II board, some further improvements on some of the initial design choices were identified. Like any design there always seems to be room for further improvements, and this is no exception. To address these further design changes I have compiled the list below;

Summary of Notes regarding Revision A

- Inductor L1 should be replaced with a 680µH, 0.125A inductor (Bourns RLB0608-681KL) which has the same footprint as the Rev.A inductor. Since the current requirements of the power supply are low enough (well below 100 mA) this is a more suitable inductor.
- Remove the 100-kΩ bleeder resistors R5, R7 and R9. They are not needed in this design as the input to the MOSFET gates are never left floating. Since the gate is always driven or held off, these resistors are not necessary. Removing them will reduce part count, layout complexity and area as well as reduce cost.
- Instead of the 1N4007 freewheel/clamping diode (D3), consider using a 1N5819 fast-acting diode which also comes in a DO-41 package. The 1N5819 features a maximum P_D = 1.25 W, I_{FSM} = 25A, andV_F of 0.6V. The maximum voltage rate of change for this device is, dV/dt = 10kV/µs superior to the 1N4007.
- Instead of the 1N5817 power supply flyback diode (D2), consider a 1N5819 which has higher V_{RRM} rating of 40V in the same DO-41 package which exceeds the needed specification for this power supply output.
- Instead of the 1N5817 diode (D1) as reverse input voltage protection, use the 1N5819 since the specs are better (V_{RRM} of 40V vs. 20V), but more importantly to reduce the number of different parts needed. Since the 1N5819 will be used in other areas as well this reduces the number of unique parts required and is available in the same DO-41 package. The 1N5819 has a slightly higher forward-voltage drop, V_F = 0.6V (opposed to the 1N5817's V_F = 0.45V) but it is still more than adequate for the job.
- Use a 1-kΩ resistor (R3, trigger input) instead of the 470-Ω resistor. This choice will still deliver the required > 2mA of current (3.8mA in this case) and if anything else will keep the different kinds of parts used to a minimum since this same value is used for R1, R4, R6, and R8.
- Use a lower voltage capacitor rating for C2. This would also be a smaller footprint. The Panasonic EEU-FR1C221 is a better choice and sufficient to meet spec with a voltage rating of 16V.
- PCB footprint for LED1-4 could be changed to the 3mm version (T-1) such as the Lumex SSL-LX3044HD (RED) and Lumex SSL-LX3044GD GREEN devices which have the same operating specs, but are available in a smaller footprint package.
- Remove JP2 (external trigger) it doesn't really seem to be needed, but instead use a jumper (2-pin) footprint just in case someone wants to solder one in.
- J1 and JP1 should be larger connectors with perhaps a wider pitch and able to accept the 18 AWG wire more easily with crimp-type terminals (the terminals and matching connectors in Error: Reference source not found page Error: Reference source not found are a bit tight).

Instead use Molex KK-style 0.156” pitch connectors which can handle 18 AWG wires preferred. Please refer to the Molex Web Catalog page 2. The Molex Series #2478 crimp terminal, #41792 header and #41695 housing looks like a good choice. There should be two separate connectors (2-pin power, 8-pin mast) placed in-line on the same edge of the PCB.

- PicKit3 doesn't work in circuit when LAMP2 MOSFET(Q2), on GP1 is inserted. This is due to the fact that GP0, GP1 and GP3 must be isolated from the application circuit for the ICSP mode to work properly. A small isolation pad or jumper could be inserted to allow these pins to be isolated easily. GP3 is already sufficiently isolated since when the trigger is not pressed only a pull-up resistor (R2) is connected to to the pin. Trace lengths should be kept to a minimum.
- PCB routing/traces should have more consideration for Power Supply layout (see the datasheet) and output connector. Minimum 2A traces with grounds going back to the supply. Separate MCU and Power grounds would be better as well and provide better isolation from power dips/surges when high-power loads such as the bell and lamps are actuated.
- PCB layout for MOSFETs should use free standing footprint versions rather than those which lay down flat. Would result in less board space and perhaps simpler routing. Maximum power dissipation, P_D = 53 mW and junction temperature $T_{\Theta j}$ = 30.77 ºC (87.39 ºF), which is 6 ºC (or 11 ºF above ambient) are well below the MOSFETs maximum ratings of, $P_{D(max)}$ = 1.38 W and $T_{\Theta j(max)}$ = 175 ºC. No heat sinking from the underlying PCB pad is needed.
- Since component count will be less and footprints will be smaller, consider keeping size of board to under 5cm x 5cm which is iTead maximum size for their least expensive production run.