Slave Unit for Digital Cameras

The built-in flashguns in digital cameras often emit a double pulse, which causes a problem when they are used with conventional slave units. ***David Gibson*** *describes an upgrade to his slave unit that aims to solve this problem.*

The emergence of digital cameras causes a problem for photographers using conventional slave units, because many digital cameras issue a double flash pulse; the main flash being preceded by a 'pre-flash' to set the exposure level or colour balance. In principle it is fairly straightforward to modify my slave unit to fire on the second pulse it receives, or to fire after a pre-determined delay. The most economical way to alter the design is to add a small microprocessor. This opens up other possibilities – for example, a 'fired' indication becomes easy to add, as does a low battery warning.

Features of new Slave Unit

A new slave unit, designed to work with digital cameras, retains all the features of my existing slave unit, but adds the following …

- 'Learning mode' matches unit to different camera characteristics.
- A low battery indication
- Adjustable firing delay for use with cascaded units, and master/slave electronic guns
- LED to indicate that the unit has been triggered (useful for setting-up purposes)

The new slave unit will be available later in the year. In the meantime, the new features are available on an upgrade board, which you can fit to a standard 'Gibson' slave unit. It should also work with Firefly units. **This article describes the upgrade board only**.

Operating Procedure

Firing the Slave Unit

The slave unit will fire in response to a sharp pulse of light from a flashgun. After firing, the red and green LEDs will light to indicate a 'status report'. (See Box: Status LEDs). After 8s the LEDs will extinguish and the unit will switch off, ready to be triggered again by the next light pulse.

The unit is programmed using the two small buttons and the two LEDs. The LEDs are only for 'debugging' – if you forget what they do you can safely ignore them! Using the Mode button is tricky too. I do not recommend that you program the unit underground – but you shouldnt need to.

Using Learn Mode

In Learn mode the unit will count pulses from a flashgun and remember how many there were. This may not always be reliable:

Specification

Run Mode Settings

Standard	fires after one pulse
Multi-Pulse	fires after a number of pulses
Delayed Fire	fires after a short delay

Learn Mode

Allows unit to be programmed to one of the above settings, or to 'learn'; from a camera or flashgun.

Flash Pulses

Minimum delay between pulses	2ms
Maximum spread of flash pulses	1s
Delayed-Fire delay	2ms *

* may be user-adjustable in future versions

Controls

Restart/Fire Button	for Manual firing
Mode/Learn Button	for programming
LEDs to show…	Operating Mode Firing Status Low Battery

Power

3.6V lithium battery on main slave board
Standby current <10uA
Battery life at least 5 years / 100,000 ops.

Status LEDs

After receiving flash pulses, the LEDs will light for 8s with the following meaning

GREEN means 'fired'

Steady Green	fired correctly
Flashing Green	fired, but extra flash pulses were received

RED means 'did not fire'

Steady Red	Learn Mode
Flashing Red	did not fire; too few flash pulses received

After Pressing the Restart/Fire button, the LEDs will light as follows

1) Brief simultaneous flash of Red and Green LEDs. This is part of a self-test feature. One action during this time is that the battery voltage is checked.

2) The red LED will blink steadily a number of times, to indicate the number of flash pulses the unit is programmed to detect. For the Standard setting this is one, of course.

Or The Red LED will give a series of double-flashes to indicate the number of ms delay in Delayed Fire setting.

3) Finally, the Green LED will light for 8s to indicate that the unit has fired.

If, during this time, the Red LED gives a very brief double-flash, repeated with 2s interval this is '**low battery**'.

At power up: When you insert a battery the red and green LEDs will blink alternately. **The unit will initialise to its Standard Setting.**

Restart / Fire Button

Allows you to:

1. **Fire the slave unit manually.** This is useful if you need to test the connection to your flashgun. The slave unit will fire when the button is *released*.
2. **Restart the operating program.** If you get muddled during programming, this button resets the microprocessor. Restarting the program like this does not change the operating mode.
3. **Check the Run settings.** When you release this button, the slave unit fires. The red and green LEDs then blink in a coded pattern to indicate the run settings. Finally, the green LED lights for 8 seconds which is the indication that the slave unit has fired (see Box: Status LEDs).
4. **Reset the operating mode**. If the Mode button is held down whilst the Restart/Fire button is pressed then the unit will reset the operating mode to Standard. For this operation, the slave unit is not fired when the Restart/Fire button is released.

Mode Button

Allows you to:

1. **Enter Learn Mode**. The Mode button is only active when the LEDs are reporting Status information so, if they are both unlit, just press Fire/Restart so that the Green LED lights. Then press Mode to enter Learn Mode, which will cause the Red LED to light.
2. **Program the number of flash pulses.** In Learn Mode: Each press of the Mode button will be counted and the unit will use this data for future firings in Run mode. Pause for unit to revert to Run mode (Red LED will extinguish).
3. **Count the pulses from a flashgun.** In Learn mode: fire your camera or flash-gun. The number of flashes will be counted and the unit will use this data for future firings in Run mode.

   The number of flash pulses the unit will count is limited only by the fact that the pulses must be spaced by at least 2ms, and that they must all happen in a one second period.

   After the unit has counted the pulses the Red LED will blink for 8s to indicate that "flash pulses were received, but the unit did not fire". When the LED goes out, the unit will be programmed and ready to work in Standard or MultiPulse mode as appropriate. You can check the setting by pressing the Restart/Fire button as described above.
4. **Set the operation to Delayed Fire.** If you do nothing at all in Learn mode it will time out after 15 seconds. The unit will be set to Delayed Fire and the LEDs will show this by blinking the status information.

the slave unit was designed specifically for caving use, to have a very high sensitivity to flash pulses but to have some immunity to caplamp beams. These two features are incompatible with the need to count an exact number of flash pulses.

You might find that the number of flashes recorded depends on the ambient lighting and the closeness of the flashgun. Therefore, as far as possible, **you should use this programming feature in similar conditions to how you expect to be using the unit underground** – i.e. do not attempt to program it in bright sunlight or with the gun very close from the sensor! *Do* make use of the verification feature (press Restart/fire and look at the Status information on the LEDs).

Note that if you do not manage to fire a flashgun within the Learn Mode timeout, the unit will set to Delayed Fire mode.

Low Voltage Detection

I have added a feature to detect a low battery voltage. As supplied, the threshold will be somewhere between 2.9V and 3.1V, which is probably OK. However, you can perform your own calibration (See Box).

Availability

To obtain a kit of parts, see **caves.org.uk / flash / newsletter.html**. Later in the year, I will providing a complete slave unit based on this upgrade.

Low Battery Calibration

You must use a fresh 3.6V lithium battery for this exercise. If in doubt, check its voltage.

1. Press the Restart button to activate the Mode button.
2. Whilst the Green LED is lit, press and hold the Mode button for at least 15s.
3. Eventually, the Green LED will give a brief double-flash. Release the mode button.
4. The battery voltage has now been recorded. The low battery threshold is set to 85% of this value, so with a 3.6V battery it will be 3.0V.

You cannot erase this value – you can only record a new one. If you record with a partially flat battery you wil move the threshold down to below 3V.

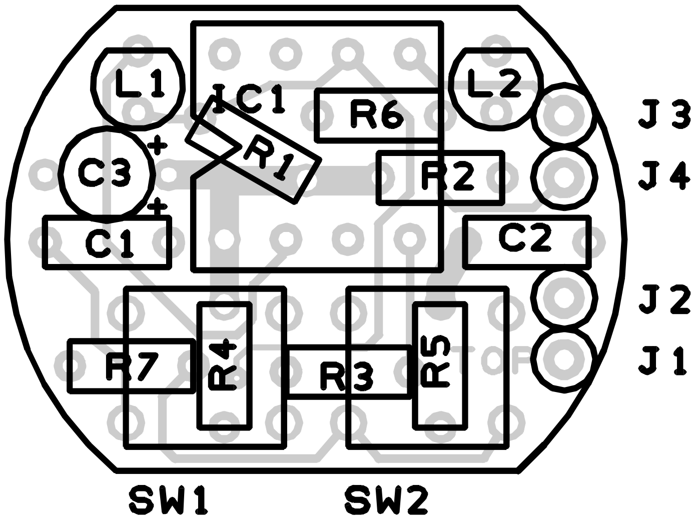
The low battery indication does not inhibit the operation of the slave unit – it is just a warning.

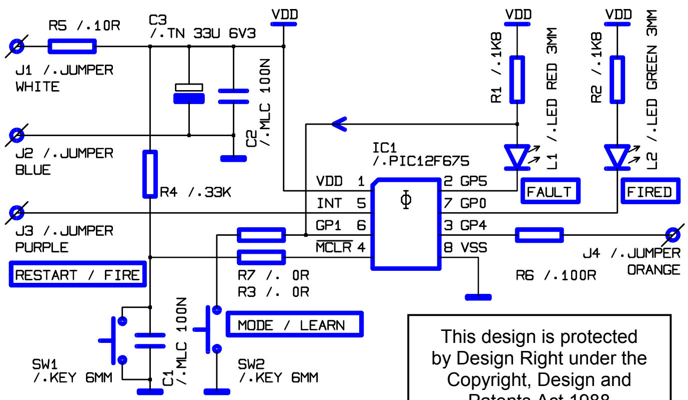
Kit Assembly

These instructions are brief. You are expected to be fully competent in soldering! Also see the assembly photos on page 24.

1. Board identification. The top side of the board has round pads; the bottom side has square pads. There is no silkscreen on the board.
2. Fit the resistors first, on the bottom side of the board, and 'top-solder' them. That is, you solder them on the same side of the board as you fitted them. Trim the leads off flush on the other side of the board.
3. Fit the IC socket before fitting IC1. The socket is not essential, but as this is a prototype, you might be expecting a software upgrade!
4. You might want to mount the LEDs so that they view sideways. It depends on how you will be mounting the unit in the container, and whether you want to view them 'in the field'.
5. Note sequence of jumpers in the diagram to right – J3 at top, not J4!
6. Fit the four jumper leads on the bottom side and secure them with a blob of silicone sealant. This is important, because otherwise the leads are likely to break off as you are fiddling with the unit.
7. If you are using my boards RALF3, SLV011 or SLV021, snip out R15 (100Ω) and remove the wires with a de-soldering iron or solder sucker. If you are using any other type of slave unit – it is up to you to work out what to do.
8. Identify appropriate pads on the solder-side of the slave unit on which to solder the four jumpers. You may wish to solder the wires so that they have to be folded through 180° to route them to the Upgrade Board. This gives you some latitude in its placement.
9. Place the board in the end of the container (if you are using my container) with an insulating sheet between it and the battery terminal. This is vitally important or you could cause the battery to explode.

Cct ref	Qty	Value		Comment
J1	1	70mm of 7/0.2 equipment wire	White	Links upgrade board to slave unit. Note that J1-J4 are not in sequence.
J2	1		Blue	
J3	1		Purple	
J4	1		Orange	
C1	1	100nF		
C2	0	100nF		***not fitted***
C3	1	33uF tantalum		Observe polarity
IC1	1	PIC12F675		Observe polarity. Use socket
	1	8 pin DIL socket		For IC1
L1	1	LED red 3mm		Observe polarity
L2	1	LED green 3mm		Observe polarity
R1, R2	2	1K8		Mount on solder-side
R3, R7	0			***fit as short circuit***
R4	1	33K		Mount on solder-side
R5	1	10R		Mount on solder-side
R6	1	100R		Mount on solder-side
SW1–2	2	keyswitch		





Circuit Description

There isnt really much to describe. The upgrade board sits between the output of the sensing op-amp on my slave unit board and the input to the opto-triac. It intercepts the pulses, counts them and decides what to do.

Pin 6 on the PIC has a dual function. As a digital input, it monitors the state of the Mode button. As an analogue input it monitors the voltage across the Red LED, which it uses as a reference for the low-battery detection feature (see later).

When the Red LED is on, the digital input buffer on pin 6 could be operating out of spec. I have not taken any action to mitigate this problem because it should not cause any malfunction.

The current through SW1 is only 100uA or so, which is below the minimum current recommended for this switch. I am not sure what the implications of this will be, but you could always drop R4 to 3k3 if you are worried that the switch will fail unexpectedly.

All of the action takes place in the software, of course. For such a simple-looking device there is a surprising amount of software – around 875 bytes, assembled from around 1500 lines of source code (yes, it is well documented!). The PIC device is one of the newer in Microchip's product range. Although it is only an 8-pin '12-series' device, it follows the '16-series' in terms of instruction set, program word size (14 bits), stack depth (8) and memory type (Flash). It also has a built-in oscillator.

The software makes use of the PIC's SLEEP function, whereby it can be turned off until it sees a pulse on pin 5. Its current during Sleep is around a microamp. There are two interrupts in use – a real-time clock that provides the timing for the various operating states, the LED flashing and so on, and the INT pin interrupt that counts the flash pulses as they are received.

I also make use of the Sleep function when the internal ADC is recording the battery voltage. A-D conversions during Sleep are much less prone to errors caused by digital noise.

The low battery detection is not as simple as it might seem, because the microprocessor does not include a voltage reference. It is not possible to use the PIC's Brownout reference voltage to detect a low battery because this uses too much current when the unit is in Standby. Instead, we use one of the LEDs as a reference. However because LEDs can have a fairly wide tolerance, this means that the unit must be calibrated. During the calibration exercise (see Box) the unit records the battery voltage relative to the LED voltage. It assumes the battery is 3.6V and subsequently uses a threshold of 85% of this, or 3V, to determine a low battery condition. Of course, this will not be exact because, at 3V, the voltage on the LED will be a bit lower too. However, the error is not critical – if the battery reaches 3.1V it is probably almost empty anyway; and the unit will still work at 2.9V.

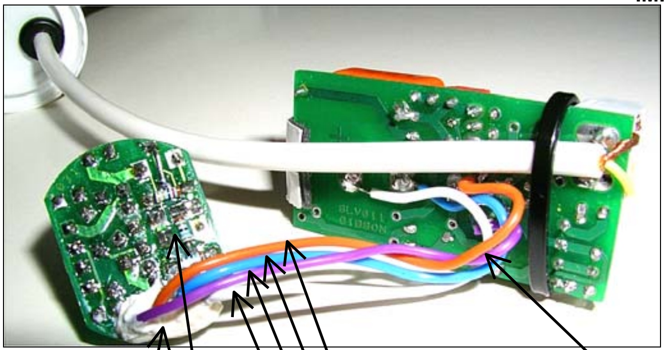
The low battery value is stored in EEPROM so it is remembered when you disconnect the battery. The other settings are not stored because it was thought that this could lead to confusion.

There are number of redundant components on the upgrade circuit board. This is because I had to design the board before I was really sure what I would need. C2 is not fitted. R3 and R7 are not needed and need to be shorted out. R1 is too low in value – the Red LED does not need to be so bright. I may correct this later.

Continued on Page 24

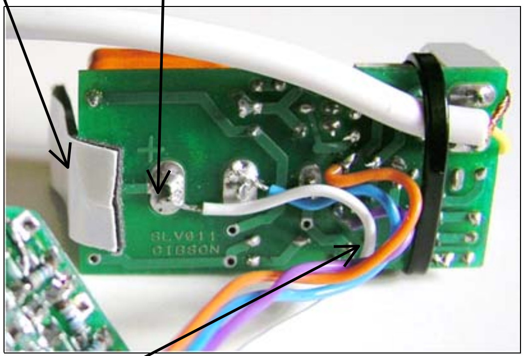
Flashgun Slave for Digital Cameras

Continued from page 17



Insulating pad over battery clip – **IMPORTANT!**

Use circuit diagram and assembly instructions of original slave unit to determine where to solder wires



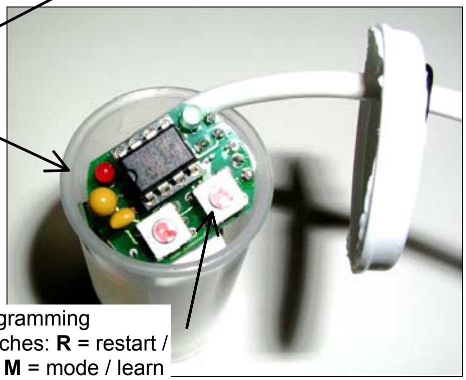
Silicone sealant as strain relief

Orange wire
Blue wire
Purple wire
White wire

Resistors on bottom of board

Fold wires to take up slack

Board fits in end of container



Programming switches: **R** = restart / fire, **M** = mode / learn

Fitting this **upgrade board** in the end of the container is not easy. You need to take **extreme care** to avoid shorting anything to the battery clip and to avoid breaking the wires. Later in the year I will be producing a version of the main circuit board that includes all the additional components (some surface mount) and this will also be available ready-built. Also under consideration is a housing that gives you easy access to the switches. At the moment I do not advise that you adjust this unit whilst caving: set it up first and then seal it to keep it dry.

CREG