

## Coin Multiplier Board User's Guide

### Description:

The "coin multiplier board" or "CMB" is a very simple board of my own design. The board shown below was built from an old board previously designed for another project unrelated to this project.

The CMB is comprised of the following components:

- PIC16F675 microcontroller: This 8-pin IC is the brain of the CMB. It contains the operating program and the data memory. The memory of the chip is non-volatile. This means that it can retain data without power indefinitely. Once you program the CMB for the correct number of output pulses, it will never forget.
- Voltage regulator circuitry: This is what allows the CMB to operate from a power supply ranging from +5V to +18Vmax. There are some filtering components that smooth the power and absorb voltage spikes. For safety, there is an onboard fuse (4A) that will only blow in the event of a catastrophic failure such as a component meltdown or the power being connected backwards. The fuse is there only to prevent a fire. Under normal circumstances, the fuse will never blow.
- Lowside Driver: The CMB uses a power mosfet transistor to control the coin pulse output line. The output is an open drain type that requires a pull-up resistor. The pullup resistor is normally on the game board but there is also one on the CMB for convenience during testing or troubleshooting of the system.
- LED: The light emitting diode (red light) is provided to indicate the operating mode, programming information, and to show when output pulses are being generated.

The CMB takes in pulses from a standard coin crediting switch and outputs crediting pulses to the coin input of the game board. The number of output pulses generated for each 1 pulse that it receives from the coin switch is programmable. For example, the CMB can be programmed to support a Canadian Loonie for games that normally accept quarters. If you drop a Loonie into a Loonie compatible coin mech., the big coin will activate the coin switch once, thereby sending a single pulse into the CMB. The CMB will then generate 4 accurately timed pulses out so that the game will grant four quarter credits. The output pulses each last 250ms and are spaced 500ms apart to give the game time to register them. Four pulses take 3 seconds to complete. If you drop more Loonies in less than that time, the CMB will not miss them. It will add 4 more credits for each coin dropped even if it is still sending out pulses. There is virtually no limit to the number of coins that you can drop one right after the other because the pulse counter is very deep. The CMB also "debounces" the coin switch using a 150ms delay. It is unlikely that anyone would drop coins into a mech faster than this period. The CMB can be user programmed to generate any number of output pulses per single input pulse within the range of 1 to 255.

### Operation:

The CMB has two operating modes, "normal" and "training". A small onboard slide switch selects the mode. The CMB also has a single LED indicator on it. The LED indicates the operating mode and provides helpful feedback to the user.

#### Normal Mode:

The slide switch should be positioned such that it is toward pin 4 of the chip U1 (toward the closest PCB edge). However, if the switch happens to be in the training position upon powerup, normal mode will run regardless. Upon powerup, the LED flashes so you can see that the CMB is alive. The number of flashes indicates how many output pulses will be generated for each input pulse. No output pulses are actually generated during this start-up phase. After the start-up phase completes, normal operation begins. For each coin you drop, the CMB will generate the exact multiple of output pulses that it is programmed for. The LED flashes once for each output pulse so you can see that it is crediting properly.

Normal mode includes one simple diagnostic feature. It can indicate a shorted coin switch. If the switch gets stuck closed (or the line is shorted to ground) for more than two seconds, the LED will blink quickly and continuously to indicate the fault. If the fault condition disappears (i.e. the switch opens again), then normal operation will resume automatically.

Filename: "Biltronix\_Coin\_Multiplier\_Board\_Users\_Guide.doc"

By: William J. Boucher, Created: Feb. 22, 2009, Revised: Feb. 22, 2009

Warning: Copyright Biltronix, All rights reserved. All information strictly confidential. For use only with permission from Biltronix. May not be modified, distributed, or sold.

### Training Mode:

The slide switch should be positioned for "normal" mode upon powerup (such that it is toward pin 4 of the chip U1 (toward the closest PCB edge). After powerup, the CMD can be programmed as follows:

1. Move the switch to the "training" mode position (such that it is toward pin 1 of the chip U1 (toward the center of the PCB). When you move the switch to the training mode position, the LED will flash quickly 10 times to indicate that it has entered training mode.
2. Manual actuate the coin switch once for each output pulse that you want the CMB to produce (when it is in normal mode). For example, if you want the CMB to generate two pulses per coin, then click the coin switch twice. If you prefer 4 pulses, then click the switch 4 times. The LED will flash once each time to click the coin switch.
3. Move the switch to the "normal" mode position. The LED will flash quickly 10 times to indicate that it has exited training mode. At this time, the new value for the number of output pulses to generate per coin is saved within the non-volatile eeprom memory of the microcontroller chip (IC U1).

### Notes:

1. If you move the switch to the training mode position and then back to the normal mode position without activating the coin switch, the previous setting for the number of pulses to generate per coin will not be changed or lost.
2. The range for the number of pulses to generate per coin is 1 to 255. Obviously, it would be completely impractical to use a very high number due to the excessive operating delays that this would cause, but suffice it to say that there is no *practical* limit to how many coin pulses that can be generated. It is highly unlikely that anyone would program the CMB for a value greater than 8 or so, but you can if you want to.

### Hook-up:

Refer to drawing "coin\_multiplier\_hookup\_02.pdf" for connection and wire color information.

To hook up the CMB in your game, you have to cut the coin switch wire and install the CMB between the two ends of the wire. You also have to connect a power source. Any DC power from +5V to +18V will make the board work but you should use the same voltage used by the game board switch input logic (most likely +5V).

You should physically mount the CMB using 4 plastic PCB standoffs or spacers using four #4 screws. The CMB is very light weight so it could be left hanging (not recommended) or just wire-tied to the existing wiring harness. Locate the coin mech switch that you want to have trigger the CMB and follow the switch wire (on the active side, not the ground side) back toward the game board until you find a convenient place to mount the CMB. Once the CMB is mounted, cut the switch wire and connect the switch-end to the CMB's switch-input terminal. Connect the game-board-end of the wire to the switch-output terminal of the CMB. Then find a convenient +5V supply wire and splice a new wire into it and run that wire to the CMB. All that's left now is to connect the CMB to ground (0V). You can find the ground return wire that comes from the coin-mech switch or any other switch that shares the ground and splice into that with a new wire and connect that to the CMB ground terminal. Alternatively, connect to any power supply or signal ground on the game board or a power regulator board. Bringing the +5V and ground leads together from the game board is highly recommended.