

# The Garage Door Closer



**February 22, 2009**

*All information on these pages, including images and program source code, is Copyright (c) 2006-2009, and cannot be used for any commercial purposes without express written permission of the author. Send an email to [The Author](#). Also, email if you have any question!*

## Competed Project



Front View

Note the hole to access R4.



Rear View



Inside

In this case, I did mount R4 and LED1 and LED2 on the bottom of the board. The height of the LEDs is determined by the height of R4. A hole is drilled in the case to allow setting R4 without opening the case

## Availability of parts and kits

Item	Price
Programmed PIC	\$4.25 US
PCB Version 1.1	\$6.00 US
PCB and Programmed PIC	\$9.50 US
PCB, board mounted parts and Sensor	\$31.00 US

- No warranty is provided. Sold “As is”. If you have a problem getting the product to function, service may be available, and a payment may be charged.
- There is no enclosure included with any parts kit.
- Wall transformer is not included.
- Above costs include shipping within the US (USPS 1st Class). No available outside of the US. [Please Email for availability and payment info.](#) Include your location, so that possible sales tax may be included. Prices are subject to change.

# Introduction

Have you ever found that someone left the garage door open? Be it a kid, a spouse, or even you! Well, that happened to me a few times, and can not only lead to thefts, but (in my own case) has resulted in snow drifts in the garage!

Over the years, I looked at various solutions, some using various home automation controllers. They worked, but each had a drawback or two. What I wanted was a simple and cheap solution that didn't rely on other things. There were also a few features that I wanted to make available right in the garage itself.

As a result, I developed the GarageDoor Closer. It's simple, cheap (only a few parts) and flexible. If the door is open, it will automatically try to close after a certain delay. It will keep trying (after a delay) until the door actually closes. The delay time is adjustable, and the unit can be disabled if you want the door open for an extended period.

The unit mounts in the garage. In my case, I mounted it close to one of the existing pushbuttons for the garage door opener. There are three connections required. The unit has a power connection, a connection to the garage opener pushbutton, and a sensor used to detect the state of the garage door. The sensor is a magnetic reed switch, with the magnet mounted on the top of the door, and the reed switch on the wall above the door. The switch I used allows a gap of up to 2 inches, so it's a bit different than the small switches used for windows and doors with alarm systems. The additional gap space is required to allow for clearance when the door is opening or closing. Mounting the sensor took me about 10 minutes, and that was using a step stool with a broken ankle!

To close the door, the GarageDoor Closer "simulates" pushing the garage door opener button. Therefore, the unit either connects in parallel with an existing door button, or a run of wire can be made back to the opener itself.

I chose to use wall power for the GarageDoor Closer, and not a battery. With changes to the PIC code, it might be possible to use "sleep" mode to conserve battery power, but the problem is that the door was left open because someone "forgot" to close it, and a dead battery would just mean that the GarageDoor Closer forgot! (Another chance of problems.)

There are two options for setting the delay time. A header and a shunt on the PC board can be used to select two different delay times, or a potentiometer can be used to set a variable delay (5 to 68 minutes). The same PIC source code is used for both, and the mode is selected by a compile time option.

Now for the real details!

## How it works

After power on initialization, the GarageDoor Closer keeps checking the status of the garage door, by means of the sensor connect to pin 4 of the PIC. If the door is open and the unit is enabled a delay before closing the door is started.

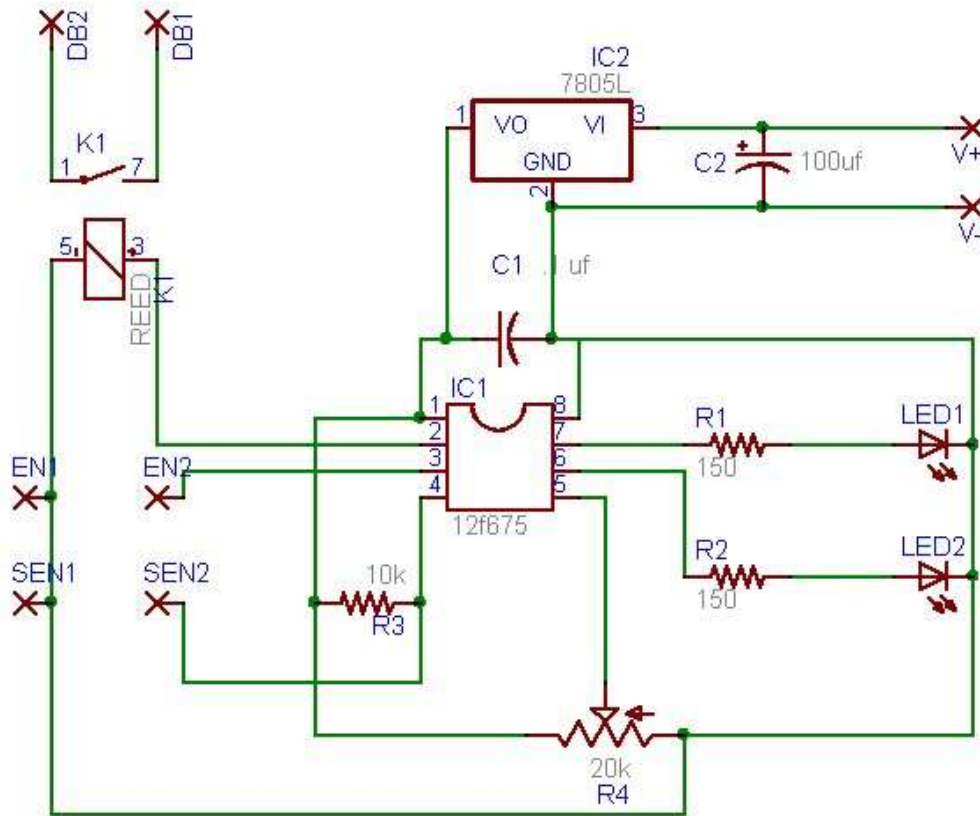
The delay is specified by reading the input on pin 5 of the pic. This is either a "shunt or no shunt" condition in the case of the fixed delays version of the firmware, or by reading an Analog to Digital value. The A to D value can be 0-1023, and using a simple shift operation in the code, this is converted to 0-63. Five minutes is added, so that a "zero delay" never occurs. That's why the unit allows 5-68 minutes for a variable delay.

When the delay expires, it uses a relay to simulate the push of the garage opener button, , by way of pin 2 of the PIC. It "pushes" the button for about a second. The delay is restarted in the event that the door doesn't close.

Two LEDs, connected to pins 6 and 7 of the PIC display various statuses during the process.

Pin 3 of the PIC is used to read the status of the enable/disable switch.

## The Schematic



Schematic

There are only a few parts needed to build the GarageDoor Closer. Please note that the schematic only includes the on-board parts and pads.

### Here's what the parts do:

- IC1 - the "brains" of this GarageDoor Closer. It's a Microchip PIC (12F675) that contains the code that really makes this project work. More detail on the code and how to program the chip will follow. For now, the interesting thing is that it's got 6 pins of digital I/O, some of which can also be used for analog input. Two of the pins (3 and 5) are used for inputs, and four will be used for output (2, 4, 6 and 7).
- IC2 - a LM7805xx voltage regulator. Using a "wall wart" type power supply of 7-12 volt DC, this will provide 5+ volts to the rest of the project
- K1 - a reed relay. When the PIC finds that it needs to close the garage door, the relay will close for about a second. The contacts of the relay are attached to your "garage door button", and simulate someone pushing that button. The pads DB1 and DB2 will be used to make this connection.

- C1 - just a decoupling capacitor.
- C2 - used to provide a bit more filtering of the supply voltage coming into the project.
- R1,R2 - current limiting resistors for LED1 and LED2
- R3 - used as a pull-up for pin 4 of the PIC 12F675. For some reason, providing this pull-up is not available on pin 4.
- R4 - used for the version of the PIC code which allows variable delay settings. This can be an on-board trimmer type pot, or can be connected to a pot on the case. (it depends on how you want to modify delays. For the fixed delay version of the PIC code, R3 can be replaced by a 2 pin header, a switch, or nothing. More about this later.
- LED1 - This is the "status" LED. For the versions I built, I used a Green LED. This part can be mounted on the PC board, or wires can be run to a case mounted LED.
- LED2 - This is the "tried close or info" LED. For the versions I built, I used a Red LED. This part can be mounted on the PC board, or wires can be run to a case mounted LED.

### **The pads are used for the following connections:**

- DB1, DB2 - used to connect to the "Door Button". This is how the GarageDoor Closer actually closes the garage door.
- V+, V- - the connection to the "wall wart" power supply
- EN1, EN2 - used to connect to an off board switch (case mounted) to enable/disable the GarageDoor Closer. It will be used to connect to SW1.
- SEN1, SEN2 - used to connect to the sensor used to see if the garage door is open or closed. It will connect to SENSOR.

DB1,DB2, SEN1 and SEN2 should be run to some type of terminal strip on the case, as it is how the GarageDoor Closer does its job.

### **Off-Board Parts**

- SENSOR - this is a reed relay with a separate magnet. Kind of like the sensors that are used by some alarm systems to see if a window or door is open. The one needed here has a 2 inch "gap space", but don't worry, it's only about \$5, and I was able to mount it in about 10 minutes, with a broken ankle. The magnet attaches to the top of the garage door (2 screws) and the relay part is attached to the wall above the magnet (when the door is closed). Here's a picture of the Sensor mounded on a garage door. Note that there's a gap of about an inch between the magnet and the reed switch.



Reed Switch on Garage Door

- SW1 - a SPST switch on the cabinet. It's used to enable/disable the auto closure.
- R4 - see above.
- LED1,LED2 - see above.
- Wall wart - one of those plug in power supplies that come with everything these days. It should provide 9-12v DC at a couple hundred MA. I just use what I got in my junk box here.

## Parts List

Part	Value
C1	.1 uf disk Cap
C2	220uf (or greater) Electrolytic Cap
IC1	12f675 Microchip PIC
IC2	78L05ACZ Voltage Regulator
K1	5v Reed Relay SIP (COTO 9001-05-00)
LED1	Green LED
LED2	Red LED
R1	150 Ohm, 1/4 watt resistor
R2	150 Ohm, 1/4 watt resistor
R3	10k Ohm, 1/4 watt resistor
R4	20k potentiometer (board or case mount - see text)
SENSOR	AMSECO 505-371G (Mouser 507-AMS-37) Magnetic Reed Switch
SW1	Panel Mount Switch
Wallwart	Wall mount transformer 7-12 v DC
SV1	6 pin header - only needed for two-layer PC board and PicKit 2 direct programming



I've found that many folks have a spare "wall wart" from something that has been broken or lost. If, however, you find you need one, one of the cheapest sources I've found is Goodwill. They go for about \$1, and you just have to find the proper one in a box or pile.

## What do the LEDs show me?

### LED1

Upon a power-up, LED1 will first blink out the version number of the code in the IC1. Its format is: *Version number - gap - Subversion number*.

For numbers 1 and greater, it will blink a series of short blinks. For zero, it will be a long blink.

So, for example, if you see 2 short blinks, a gap, and a long blink, it will indicate that the firmware is version 2.0.

Following this, the GarageDoor Closer will be running. LED1 will start displaying status. It will be in one of four states:

- Always on - the garage door is closed and the GarageDoor closer is enabled
- Always Off - the garage door is closed and the GarageDoor closer is disabled
- Fast Blink - the garage door is open, and the GarageDoor Closer is enabled and is waiting to close the door
- Slow Blink - the garage door is open and the GarageDoor Closer is disabled

### LED2

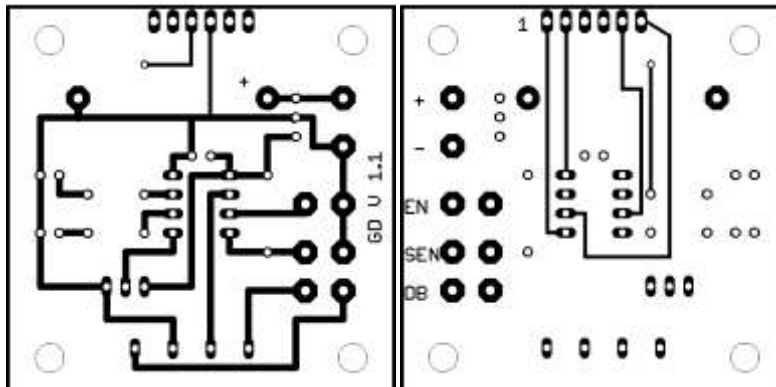
LED2 is used to display additional information. It will be in one of three states:

- Always on - the garage door is open and the GarageDoor Closer has tried to close the door. You'll see this if, for example, something is blocking the door and it won't close
- Always off - nothing interesting has happened
- Blinking - the current delay setting is being reported. This will occur when the GarageDoor Closer first detects that the door is open. As with the version display, a long blink indicates a zero, while a series of short blinks indicate a non-zero number. "short short short gap long" would indicate a 30 minute delay. The hundred digits will only be displayed if it's not zero.

## Building it

The GarageDoor Closer can be built on a perf board, or on a PC board. Here's an image of the copper side of the PC board. The image needs to be converted into a 2 inch by 2 inch image if you plan on making your own. See the "parts availability" section if you like to buy a PCB.

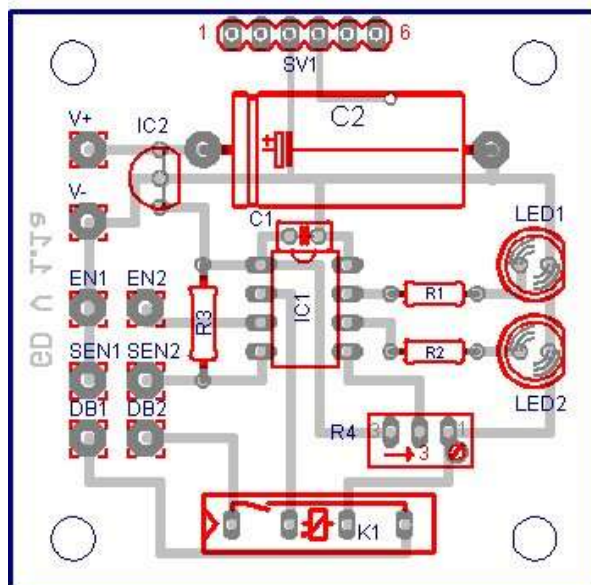
The Bottom and Top sides of the PCB.



The top side of the board - only needed if you'd like direct programmability from a Microchip PicKit 2 programmer.

If you use the PC board, here is the parts layout. The things to consider are:

- Pin 1 of the LM7805 (IC2) is towards the bottom of the board. The flat side of a TO92 case will be towards C2.
- R4 can be located off the board. If the fixed delay version of the firmware is used, the 2-pin header or switch will be connected to pins 1 and 2.
- LED1 and LED2 can be located off board
- You may want to consider using a socket for IC1. That will make it easier to install a new PIC with updated firmware.



(Note: only bottom traces are shown. Parts mount on top.)

# The Code for the PIC

The code that's stored in the PIC is fairly straightforward. There's an Interrupt Service Routine (ISR) that handles the clock (Timer0) interrupts, and the "main" code.

The ISR keeps track of time by "ticks" which results in a ten second and one minute counter. See the code itself for more info.

The "main" code just loops and checks the status of the sensor and the enable/disable switch, and acts accordingly.

There are a few options available in the code to control the generated file in the PIC:

- `#define ANALOGDELAY` (defined by default)

If defined, the delays are set by using a variable register for R4. If not, two fixed delays can be selected with a jumper. To specify the two delays, the following defines are used:

- `#define MINUTEOPEN0 0x0f` ; minutes open before close (jumper open) (value is in hex)
- `#define MINUTEOPEN1 0x3c` ;second value (jumper closed) (value is in hex).
- `#define SHORTDELAY` (defined by default)
  - If defined, R4 allows delays between 5 and 68 minutes. If not defined, R4 allows 5 and 132 minutes.
- The following is used to select if SENSOR is normally open or normally closed when the garage door is closed.
  - `#define TESTSENSOR btfss` ;nc (defined by default)
  - `;#define TESTSENSOR btfsc` ;no

Any other questions, please look at the code. Please be careful if you change other things!

Here is the PIC code for this project: [GD-1.2.ZIP file containing GD.ASM GD.HEX Version 1.2](#)

## **Additional author notes**

**7/28/2006**

After using this for a few months, I'll pass on my view of various options. First of all, using a pot to set the delay time does make the most sense. Some folks may like the jumper version, but it's a bit of a pain to burn new PICS if you want to change the delay.

Also, I've found that an internal/ PCB mounted pot works best. I tried an external pot version with an external knob, and it kept changing delays. This was due to the knob getting bumped, by yours truly!. Anyway, with a board mounted trimmer pot, and a small hole in the case, the delay can be adjusted as needed, but it won't get changed by mistake.

**11/05/2006**

I got a report back from someone who got the PCB and pre-programmed PIC. All went well, and it's installed and running. If you've built this, please email me with your thoughts!

**12/01/06**

I got a shipment of PC boards, so I have a good stock right now. (Version 1.1 boards)

Version 1.2 of the firmware is available. The only real change is a #define to allow for longer delays. The Zip file also now contains the .asm file as well as the .hex.

\*\*\*\*\* NOTE: If you are using the Pickit 2 interface port, the sensor contracts must be open during programming!

**12/01/08**

Using R4 from the parts list, clockwise increases the delay. One full turn is about 2 minutes with SHORTDELAY defined in the code (default).

**02/21/09**

Version 1.3 of firmware – changes to the “blink” method for the LEDs. Beta only.