Tesla Battery Charge Reminder

I built this project because my wife and I have forgotten to plug in our Tesla several times after parking it in the garage (I forget, she just doesn't do it). It usually happens when we unload groceries because I wait to plug in until after the bags are out of the trunk, otherwise the charging cable is a trip hazard and the charge plug could get bumped. It's a tight fit in our little cluttered garage. After we're done unloading, I'm supposed to plug in the charger (it's a "blue" task) but occasionally I forget. This has forced us to use the gasoline truck the next morning if we need to go out.

Our home is in the country with no Superchargers nearby so it's important to charge up every day. We tried setting a reminder alarm for 8PM every evening but it eventually became background noise. I would remember plugging in the car but it turned out to be yesterday, not today. The obvious solution is for me to improve my memory but I'm afraid it's going the other direction.

I wish Tesla would add some code to their phone app to remind me to plug in the car. Their app knows the car is home, not charging, and the state of charge. It seems trivial for their app to send out a reminder. This issue has been discussed numerous times on the Tesla forums but nothing has been done. There is a third party "Stats" app that can do the task and a whole lot of other stuff but it costs 50 dollars.

I don't know how to program phone apps but I can hack together some code that runs on an Arduino. Since only 5 I/O signals are needed, an ATtiny85 microcontroller can do the job although an Arduino Nano could also be used (I've provided code for both). I have a lot of spare electronic parts so I based the design on the components I already had in stock. This project ended up costing me nothing but time.

Two sensors are necessary, one to tell if the car is in the garage and the other to tell if the charge cable is hanging on the wall hook. I used an HC-SR04 Ultrasonic distance sensor to tell if the car is in the garage otherwise it reads the large distance over to the opposite wall. The second sensor is a push button switch at the base of the charge cable wall hook. If the switch is depressed, then the cable is hanging on the wall and not plugged into the car.

Obviously you can defeat the system if the charge cables are not on the wall hook and not plugged into the car. I'm not worried about this situation because I would never leave the charge cables in a heap on the floor. My solution also requires that if the car is in the garage, it must always be attached to the charger. That's our normal mode so we don't gradually loose charge to the "Vampire effect".

Circuit Description:

A simple active buzzer is turned on by the microcontroller if the car has been in the garage for over 30 minutes and the charge cord is still resting on the wall hook. This gives me plenty of time to unload the vehicle. As a debug aid and a power indicator, the LED blinks to show how many feet the ultrasonic sensor is measuring.

The circuit runs on less than 100ma at 5 volts DC which is provided from a spare wall wart power supply. When power is applied, the code turns on the buzzer for 1 second to confirm it works.

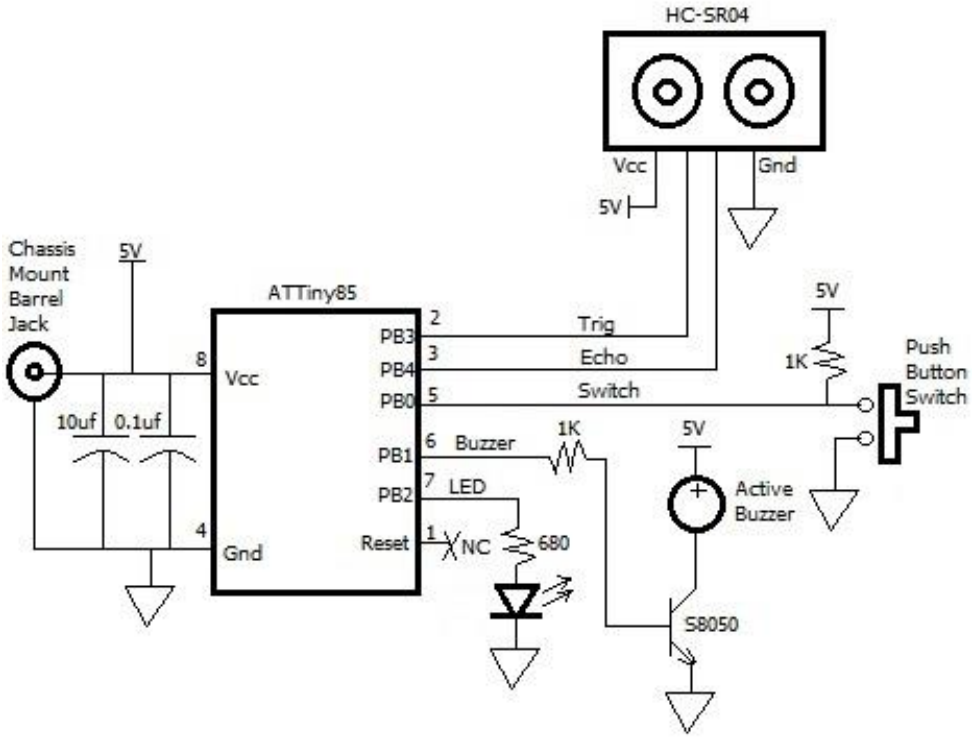
The main loop, which runs every 10 seconds, reads the push button switch on the wall hook and the HC-SR04 distance measurement. If the car is in the garage and the charge cable is still on the hook, a counter is incremented. Once the counter reaches 30 minutes, the buzzer is turned on.

Two conditions will zero the counter; if the charge cable is taken off the wall hook or if two consecutive distance readings show the car has left the garage. I was worried that the HC-SR04 might occasionally give a bad (large) distance reading which would zero the counter so I added the requirement for two consecutive readings.

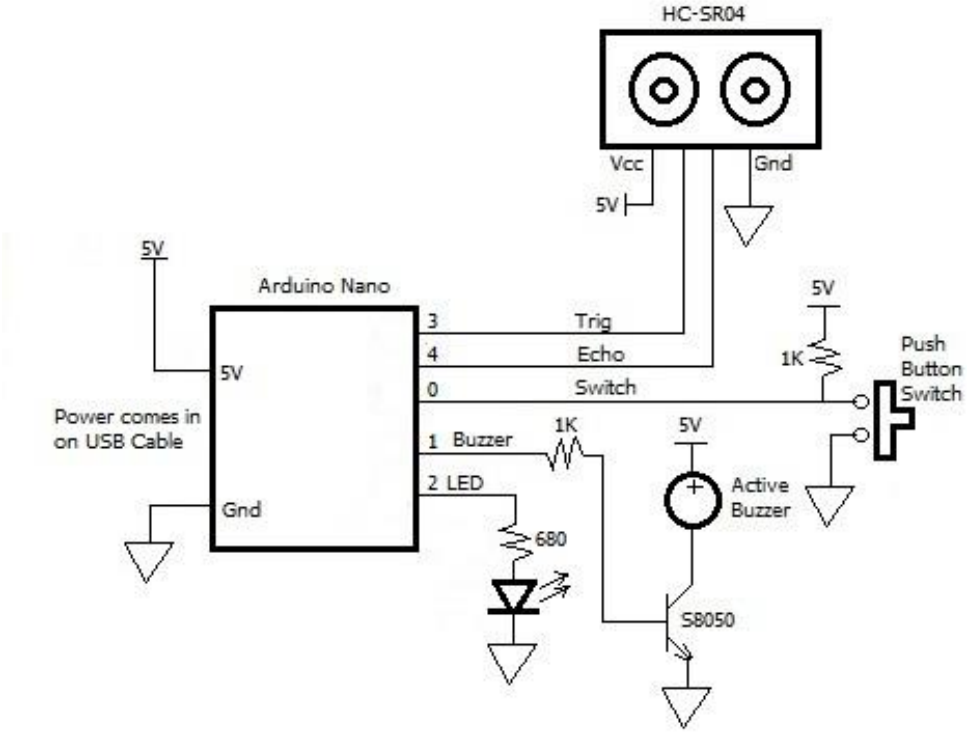
Blinking the LED to show the distance takes 10 seconds and is done at the end of each loop to control the repeat rate. If the distance is under 1 foot, no blinks are given for 10 seconds. If the distance is 10 or more feet, the LED is turned on for 5 seconds and turned off for 5 seconds. For a distance of 1 to 9 feet, the LED is turned on and off at a 1 second rate from 1 to 9 times. Then the LED is left off until the total time has reached 10 seconds.

The code for the ATTiny85 is called "Attiny_Buzzer.ino" and the code for a Nano is called "Arduino_Buzzer.ino". Both are available at my GitHub repo. The code is the same for both, just different comments.

Schematic for ATTiny85:



Schematic for Arduino Nano:



Parts List:

ATTiny85V or Arduino Nano (see note below)		$2 or $21
8 pin DIP socket for the ATTiny		$0.45
Chassis mount barrel jack for the ATTiny		$2
HC-SR04		$4.45
Push Button Switch		$0.5
S8050 PNP Transistor		$0.38
Active Buzzer		$1
5 Volt power supply	Wall Wart or USB	$9
Plastic project box		$5
General purpose proto board or OSH Park PCB for Tiny		$5
V115 Series N208-736 Corner Brace		$9
Qty 2 1k resistors		$0.50
680 ohm resistor		$0.25
4ma LED		$1
10uf Cap for the ATTiny		$0.50
0.1uf Cap for the ATTiny		$0.50

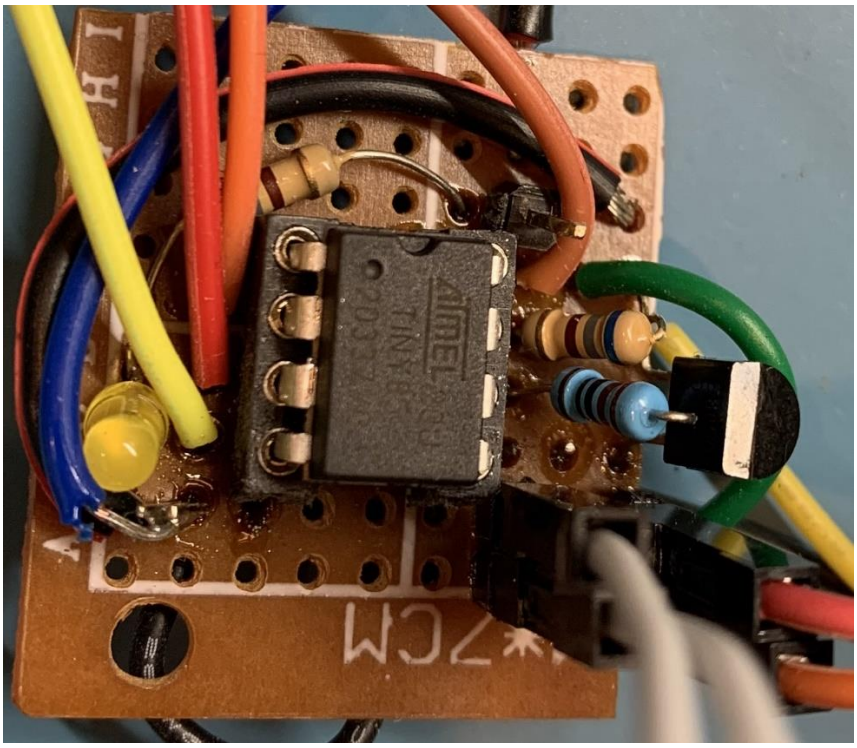
Notes:

An alternative to the ATTiny85 is an Arduino Nano. A Nano with a USB interface will eliminate the need for a separate programmer. The ATTiny code is the same as the Nano code.

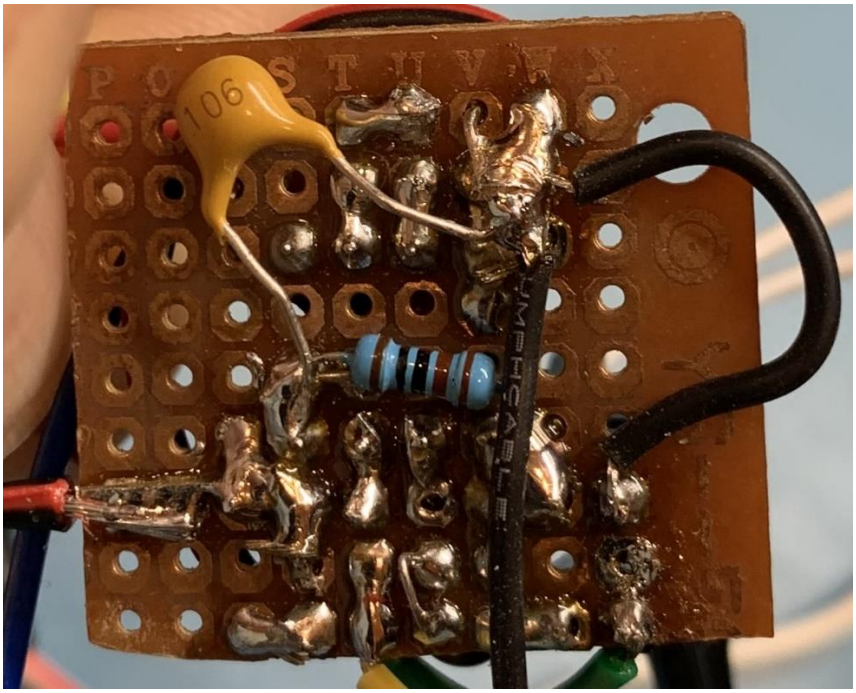
A programmer like the Tiny AVR is needed for the ATTiny85 but not the Nano. It costs $19

Assembled Circuit

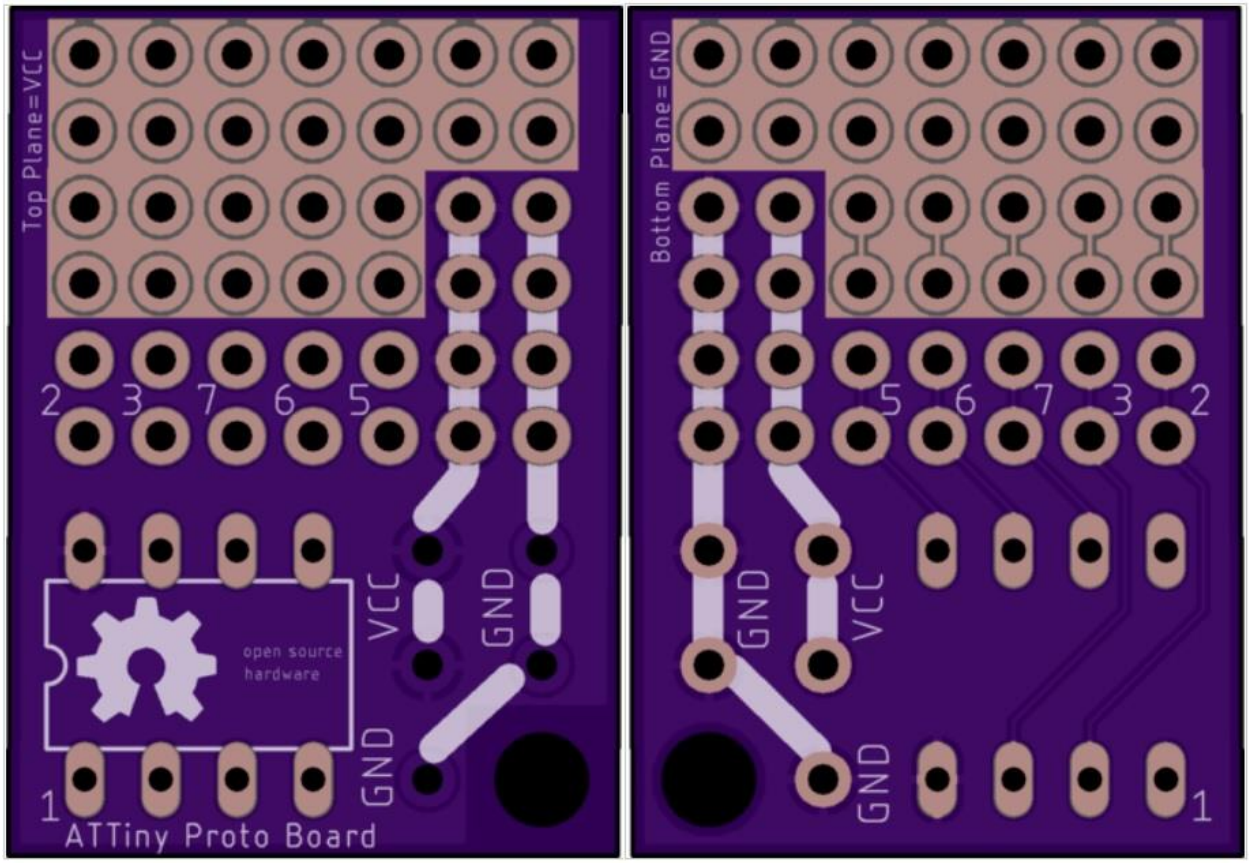
Front Side – 8 Pin DIP socket for ATTiny85



Back Side Solder – the bigger the blob, the better the job



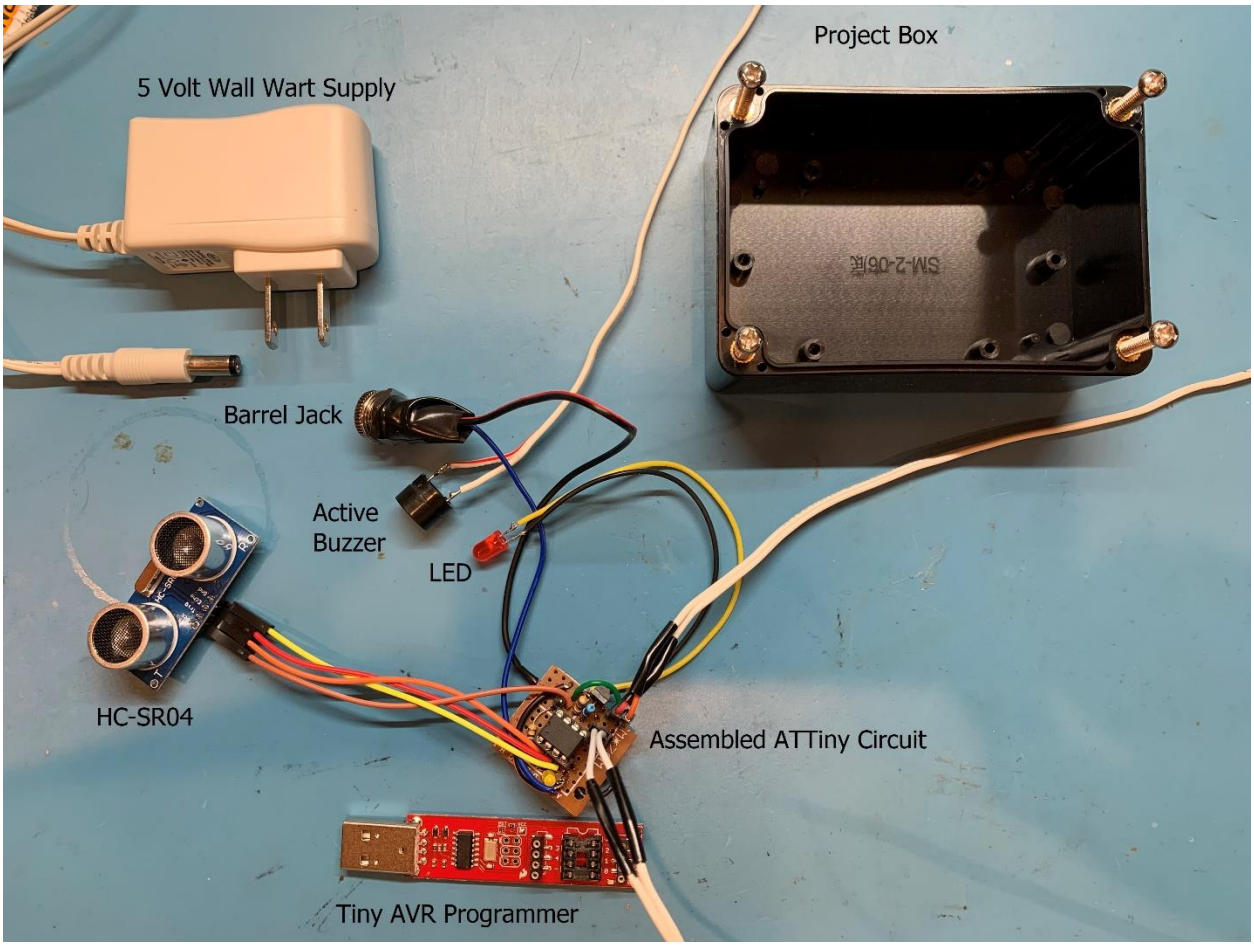
Optional – The OSH Park PCB shown below for the ATTiny85 can be used instead of the “ugly wiring” method. It is a general purpose 20.4mm x 28.2mm prototyping board. The board will still require a few jumpers but it’s much easier to assemble than a blank board. Install an 8 pin DIP socket for the ATTiny85. The Eagle board file “ATTiny85.brd” or the Gerber layout “ATTiny85_2021-12-29.zip” can be downloaded from my repo for fabrication at OSH Park ($4.45 for 3 boards) or elsewhere. This board does not provide programming connections so you’ll still need a separate programmer like a Tiny AVR.



Top View Bottom View

If using the Arduino Nano, no PCB is needed.

Prior to packaging



An 8 pin DIP socket allows for moving the ATTiny85 from the Tiny AVR USB programmer to the assembled circuit board.

Assembled Project Box

Holes were drilled in the project box for the HC-SR04 transmitter and receiver sensors as well as the LED. They were epoxied to the box with JB Weld. The barrel jack power connector hole was drilled on the side. The lid for the project box was not used because the box was screwed directly to the mounting board.

Charge Cord Wall Hook with Switch

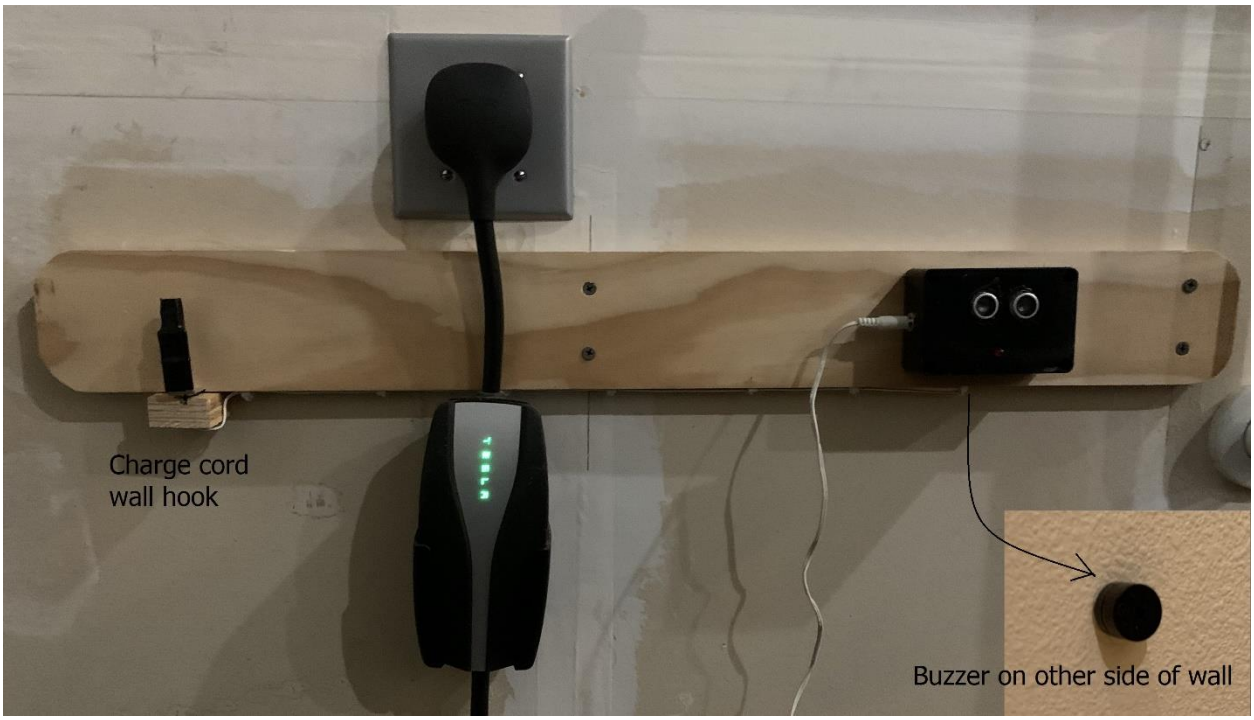
Popcicle sticks with electrical tape wrapped on corner brace (see below)

Push button switch

Wood block for support

National Hardware
V115 Series N208-736
Corner Brace

Finished Project



The mounting board is screwed to the wall studs for support. A large hole was drilled in the board directly behind the project box for wire routing. Wires from the switch under the wall hook are tacked under the board with staples and fed into the project box. To make sure the buzzer could be heard in the house, it was located on the other side of the wall, and glued in place. Two small holes were drilled in the wallboard for the wire connections.