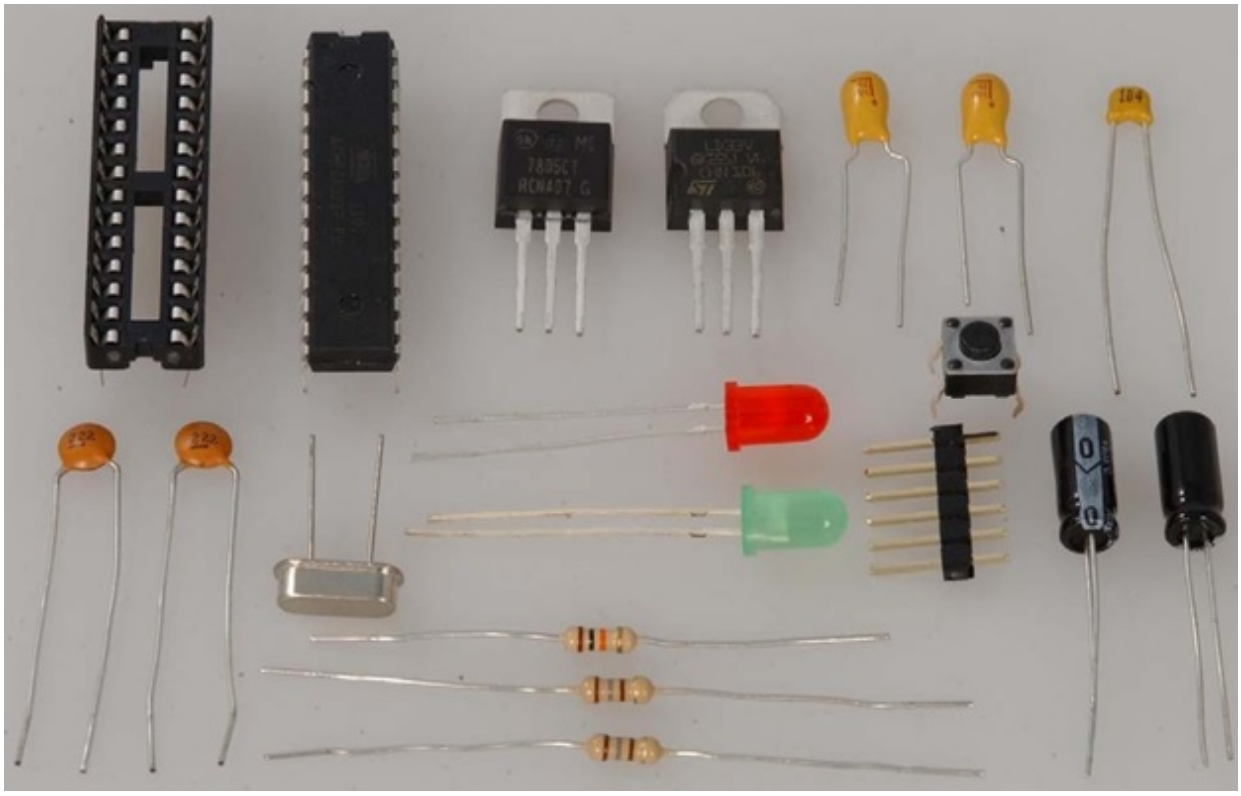
Arduino on a Breadboard
by Ryan Winters

There are many reasons to want an Arduino circuit on a breadboard or PCB. Building the Arduino-compatible circuit on the breadboard or PCB takes up much less space than the standard Arduino board. Some projects don't always require every pin to be used on the I/O headers, or maybe you won't be using a shield, but you still want the brain of the Arduino at the heart of your project. The following steps will outline how to assemble the circuit on a breadboard. I am borrowing a majority of the walk-through from the Arduino site.



Kit Components:

2129334	1	IC, ATmega328P
526248	1	Socket, IC, 28-pin, 0.3"
51262	1	IC, 5V regulator, 7805T
334035	1	LED, Red, 660nm, T1-3/4
693901	1	LED, Green, 565nm, T1-3/4
691104	1	Resistor, 1/4W, 10KΩ
690689	2	Resistor, 1/4W, 180Ω
29891	2	Capacitor, radial, 10µF, 50V
325139	1	Crystal, 16 MHz, low-profile
15405	2	Capacitor, ceramic disc, 22pF, 50V
15270	1	Capacitor, ceramic disc, 0.1µF, 50V

153251	1	Switch, pushbutton, OFF-(ON)
153700	1	Header, 6-pin, 1 row, vertical, 0.1"
242114	1	IC, 3.3V regulator, LM1117T-3.3
94078	2	Capacitor, tantalum, 10µF, 25V

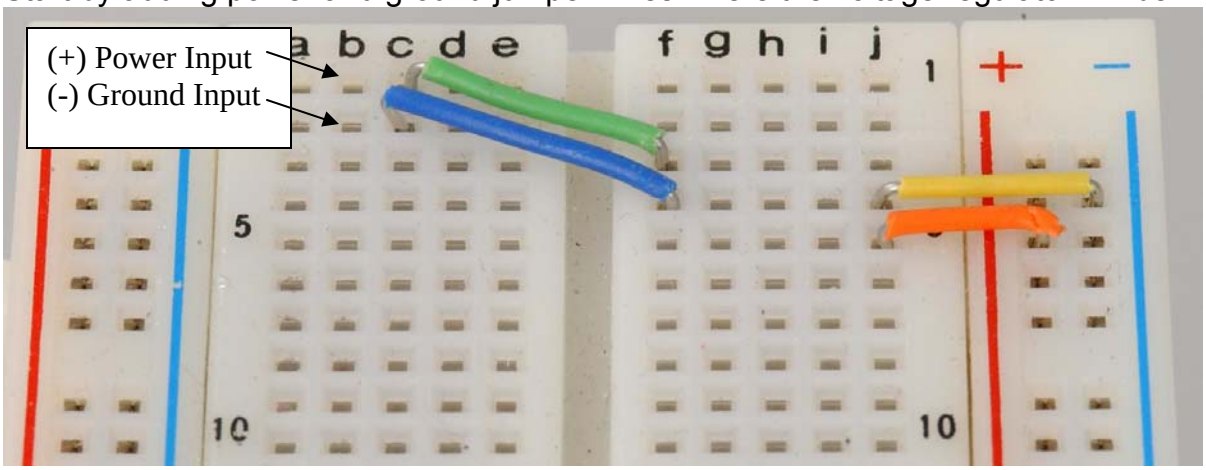
Other components you may need:

36768	1	Wire, Hook-up, 22AWG solid, 100', Blue
2127718	1	Wire Jumper kit, 22AWG, 70pcs, 14 lengths, pre-stripped
20723	1	Breadboard, 830 point, 6.5" x 2.125"
2117341	1	FTDI Breakout board, 5V, USB to Serial
252786	1	Power Supply, Wall adapter, 9V @ 1.2A
281851	1	DC Power Jack, 2.1mm
2128067	1	Battery holder w/ cover & switch, 9V, 6" wires
198731	1	Battery, Energizer 9V

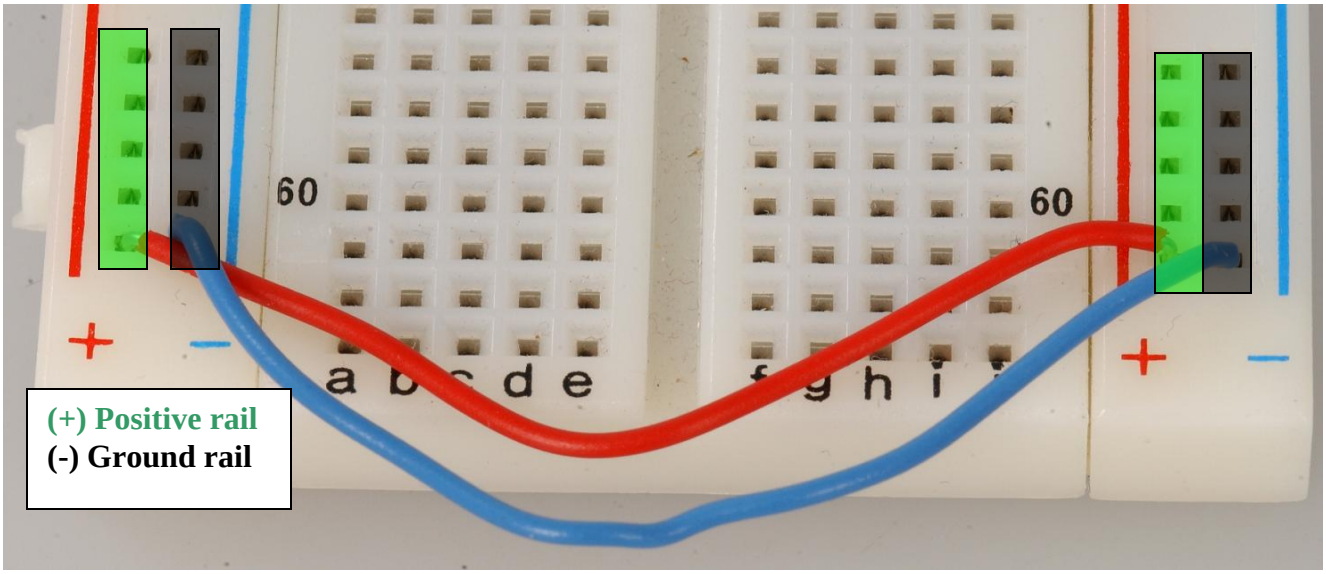
Add components for the power supply

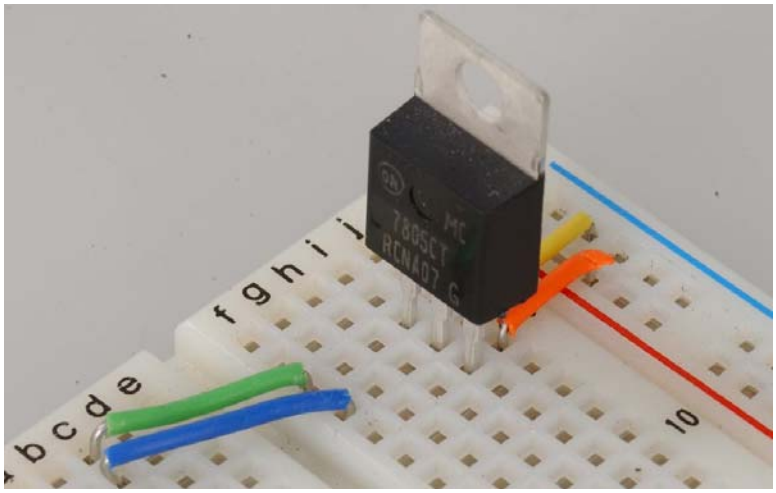
The Arduino power jack can accept an input voltage of 7-16 volts, but the most common input source is a trusty 9V battery. Because most sensors and chips require a 5V source, we will need the 7805T voltage regulator (P/N: 51262) to cut the 9V down to a component friendly 5V. If you use an input power source of less than 7V, you will not get 5V from the regulator. If you connect more than 16V, you risk damaging the IC. A 9V battery or 9-12VDC power supply is reasonable.

Start by adding power and ground jumper wires where the voltage regulator will be.



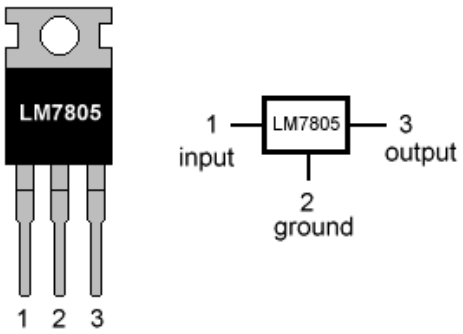
Add connecting power and ground wires at the bottom of the breadboard to connect the ground rails together and the power rails together.



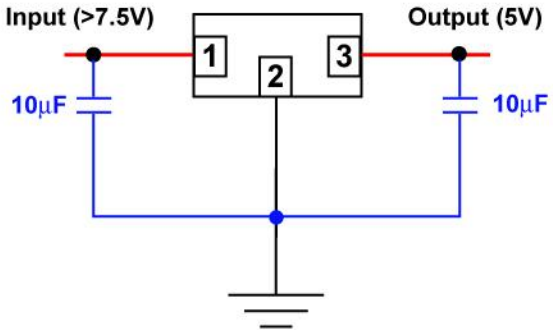


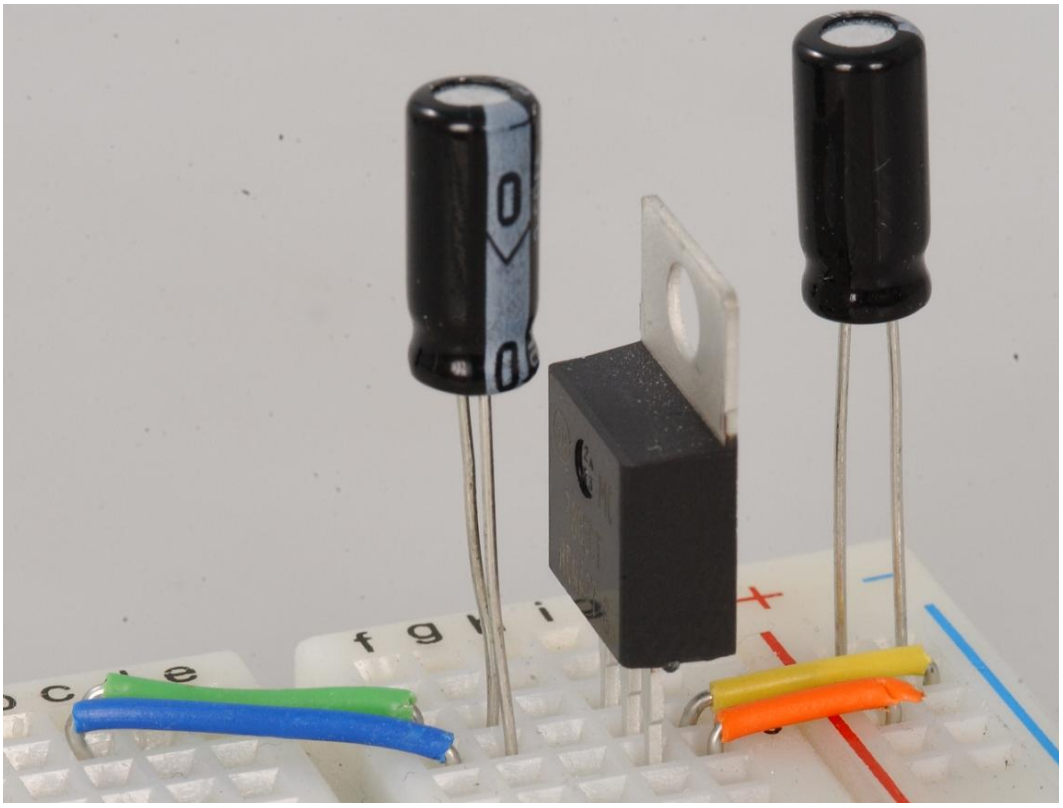
The 7805 voltage regulator is a TO-220 package, so if you have the component facing you with the leads pointing down, the first pin (left side) is where the positive input from the external power source or 9V battery will connect. The middle pin is ground (negative), and the third pin (right side) is the 5V output side. If you haven't already, add wires to connect the output side of the regulator to the power rail of the breadboard and the ground to the ground rail. It is also necessary to clean up the power by adding the 10µF decoupling capacitors between the input power and ground and also on the output side between the power rail and the ground rail. The capacitors are polarized, so the negative side is marked with a minus (-) sign. The negative side goes to ground, and the other pin goes to the positive voltage.

LM7805 PINOUT DIAGRAM

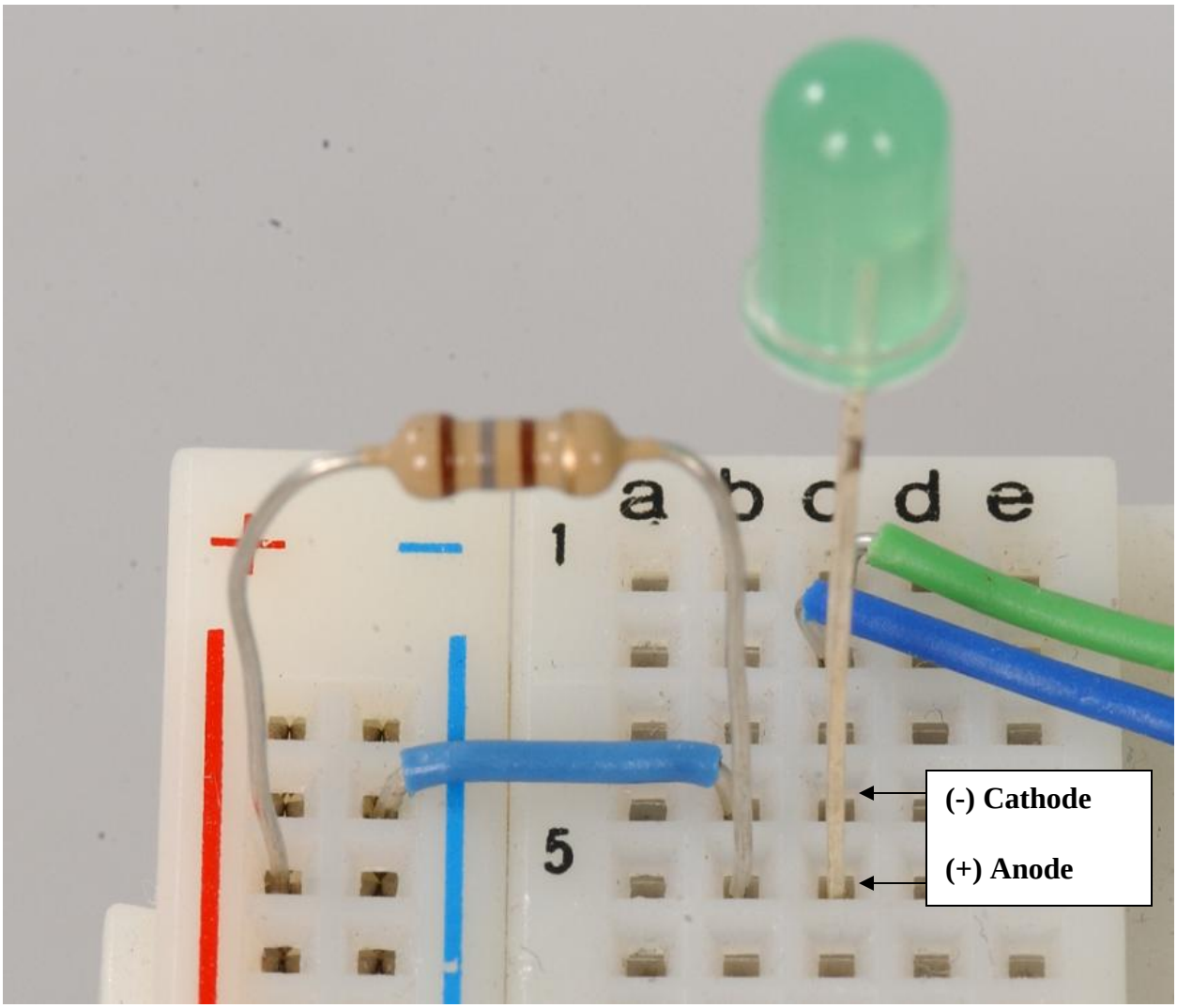


LM7805

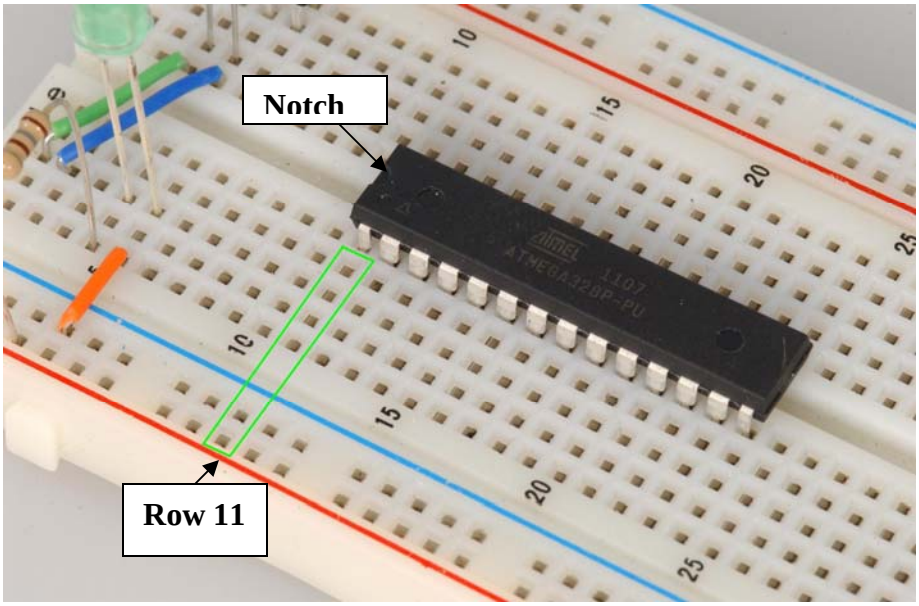




We are now going to add a power LED. This is helpful mostly to let you know the board is getting power. You can use the green or red LED. I chose the green LED because it tells me the board is good to go. The red LED I will use as the Arduino pin-13 LED. When the circuit is complete, I will load the sample "Blink" sketch to verify everything is working. Place the LED close to the input source and at the top of the breadboard. Connect a jumper wire from the negative lead (short leg) of the LED to the ground rail, and install a 180Ω resistor from the positive lead (long leg) of the LED to the power rail.



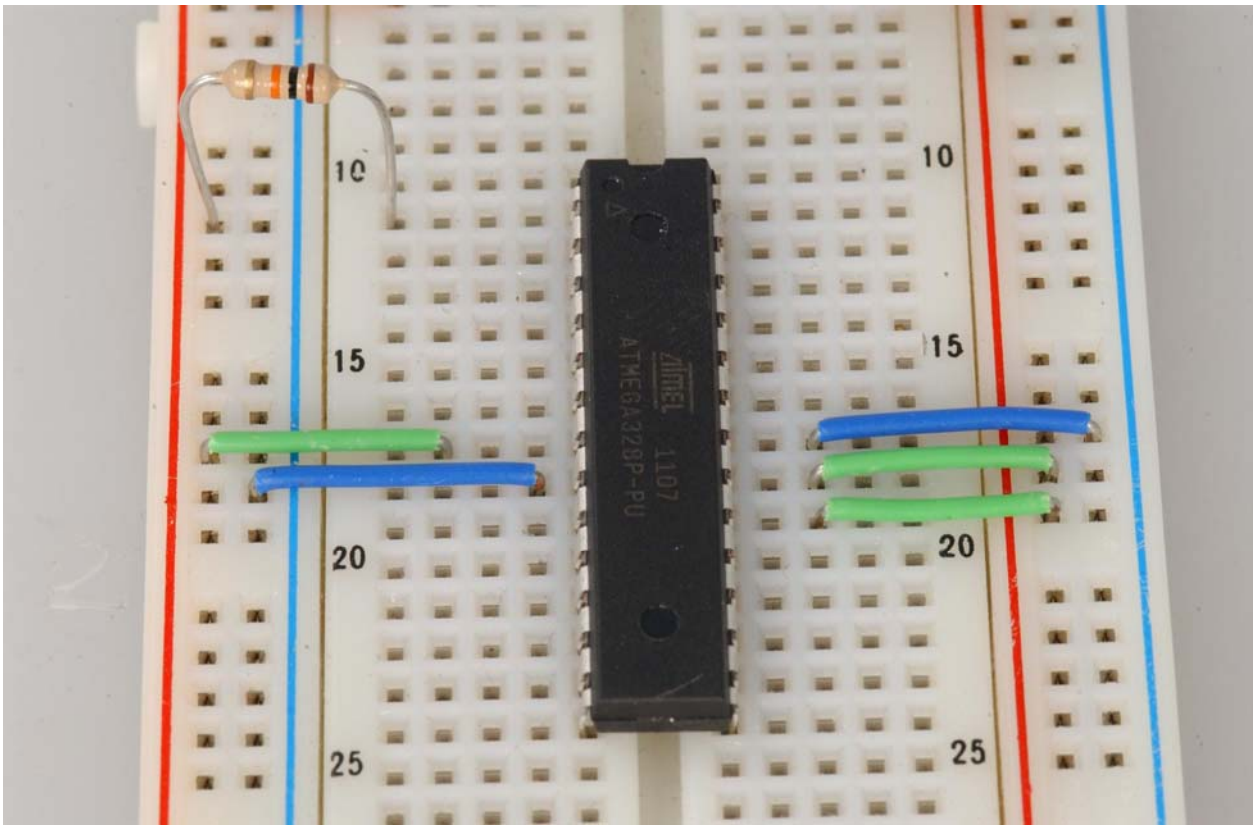
Install the ATmega328 chip so the notched side of the IC is at the top. If you are mounting the components on a PCB, it is a good idea to use the socket. If something happens to the IC it can be easily replaced. This means pin 1 of the Arduino will be on the left side. I placed the IC so that pin 1 was on row 11 of the



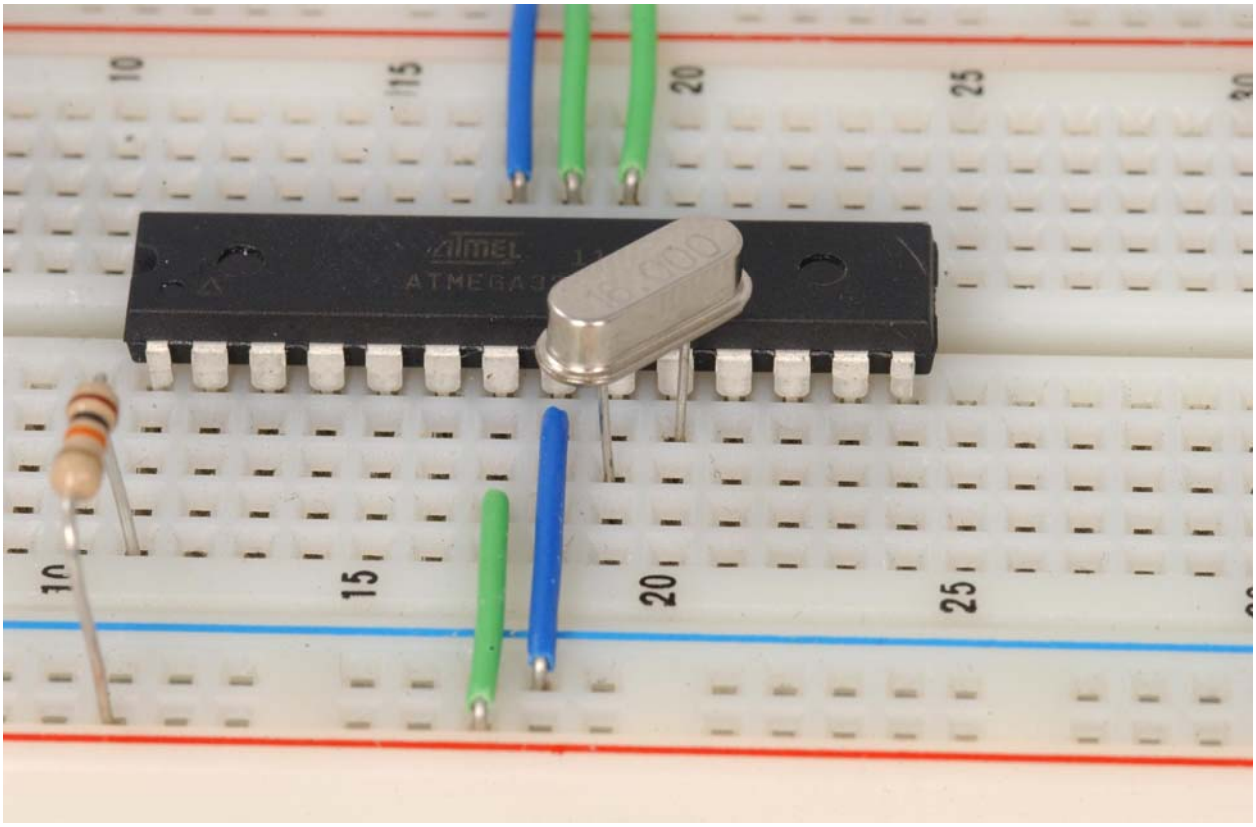
breadboard. This made it easier for me to count down when looking for other pins. Add the 10KΩ pull-up resistor to the +5V rail and connect the other end to the RESET pin on the ATmega328 (pin 1). This will prevent the chip from resetting itself during normal operation. We will be adding a reset button later. Now add jumpers for power and ground for the following pins.

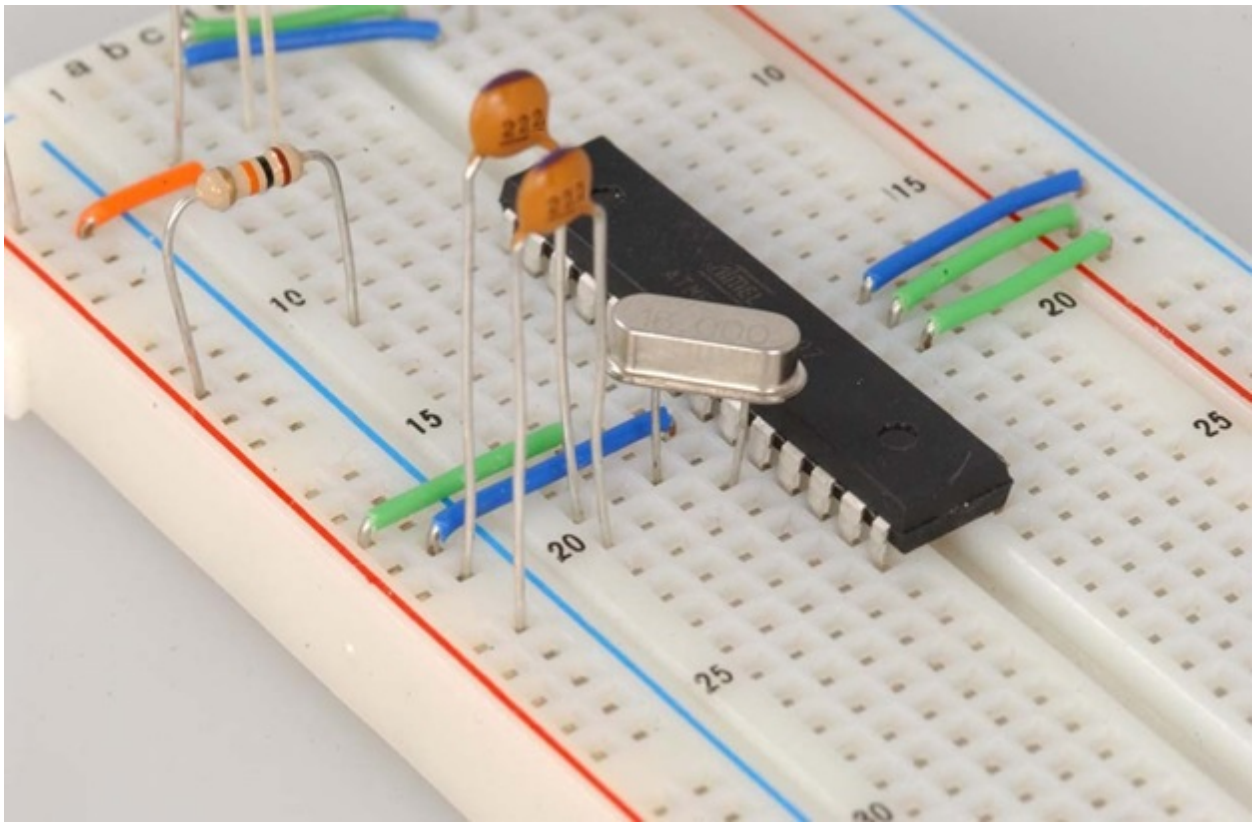
•**Pin 7** - VCC, digital supply voltage (+5V)
•**Pin 8** - GND (ground rail)
•**Pin 22** - GND (ground rail)
•**Pin 21** - AREF, analog reference pin for ADC (+5V)
•**PIN 20** - AVcc, supply voltage for the ADC (+5V).

Pin 20 needs to be connected to power if ADC isn't being used, and if it is, it needs to be connected to power via a low-pass filter. (A low-pass filter is a circuit that lessens noise from the power source.)

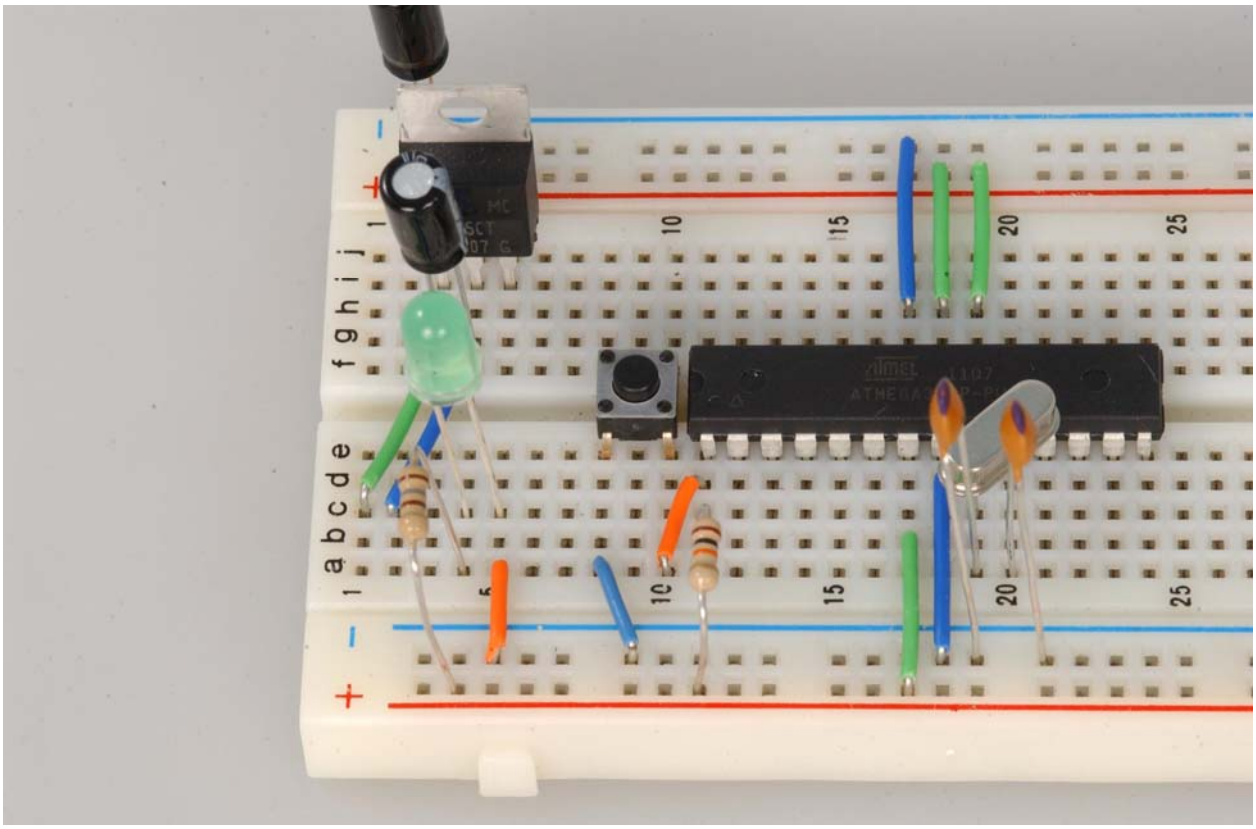


Add the 16 MHz external crystal between pin 9 and pin 10 of the ATmega328. Then add one 22 pF capacitor from pin 9 to the ground rail, and the other 22 pF capacitor from pin 10 to the ground rail. See below.


ATMEL
ATMEGA
16.000
10
15
20
25


a b c
10
15
20
25

Add the momentary button as a reset switch for the chip. There should be room just above the ATmega328. Install the switch so it spans the gap on the breadboard the same way the IC does. Add a small jumper wire from pin 1 of the ATmega328 to the bottom leg of the pushbutton; it should be the pin closest to the IC. Add another jumper wire from the top left leg of the pushbutton over to ground.

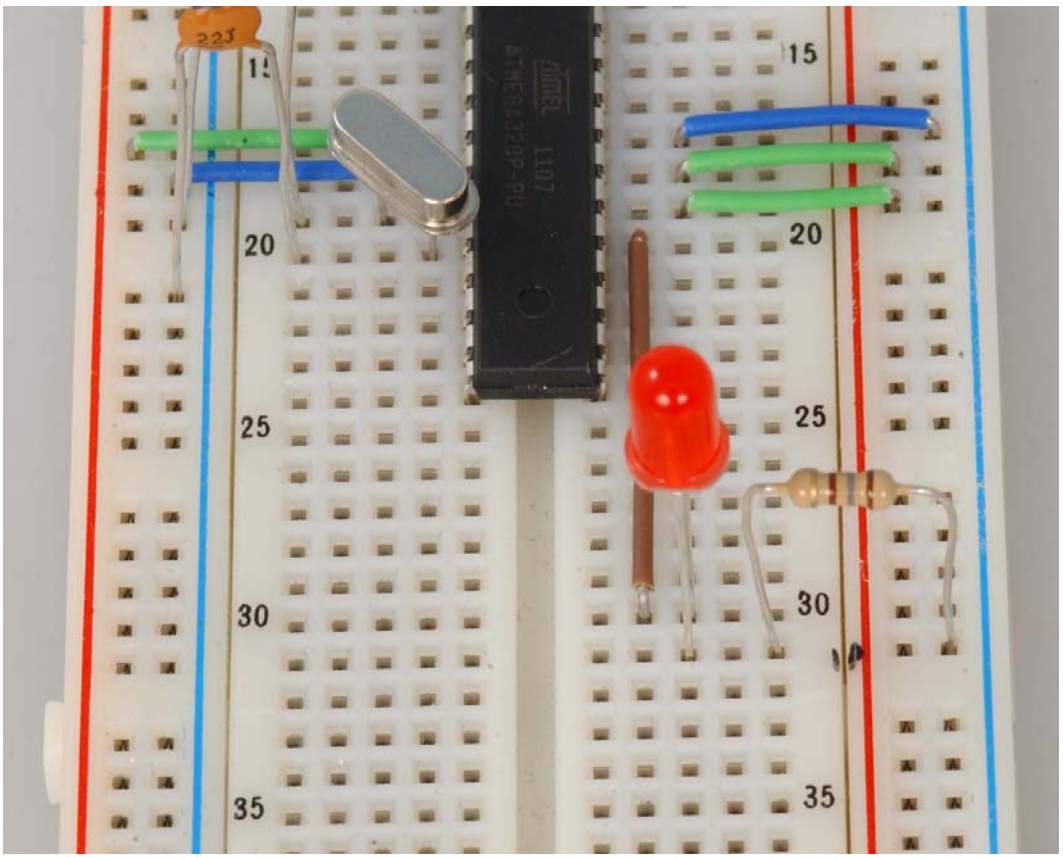


Now we will add the Arduino pin 13 LED. **Note:** Pin 13 on the Arduino is not the same pin 13 on the ATmega328 IC. Pin 19 on the IC is actually the pin for Digital pin 13 on the Arduino. If you are unsure or just want to see the pin out for the ATmega328 IC, refer to the diagram below, or you can view the short form datasheet for more complete specs. A pin out of the ATmega328 is also shown below. (A longer version of the datasheet is also available.) Place the LED below the other components on the breadboard. Connect a jumper wire from pin 19 of the microcontroller to the cathode (longer lead) of the LED. Use the remaining 180Ω resistor to connect the anode (short lead) of the LED to the ground rail.

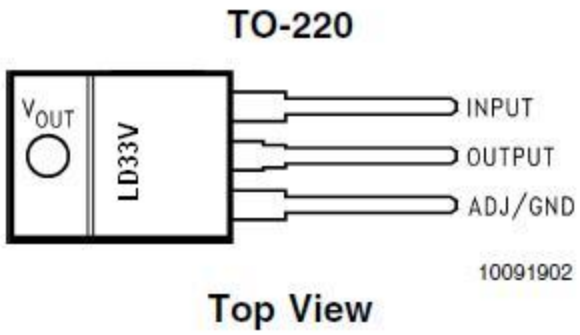
ATmega328 Pin Mapping

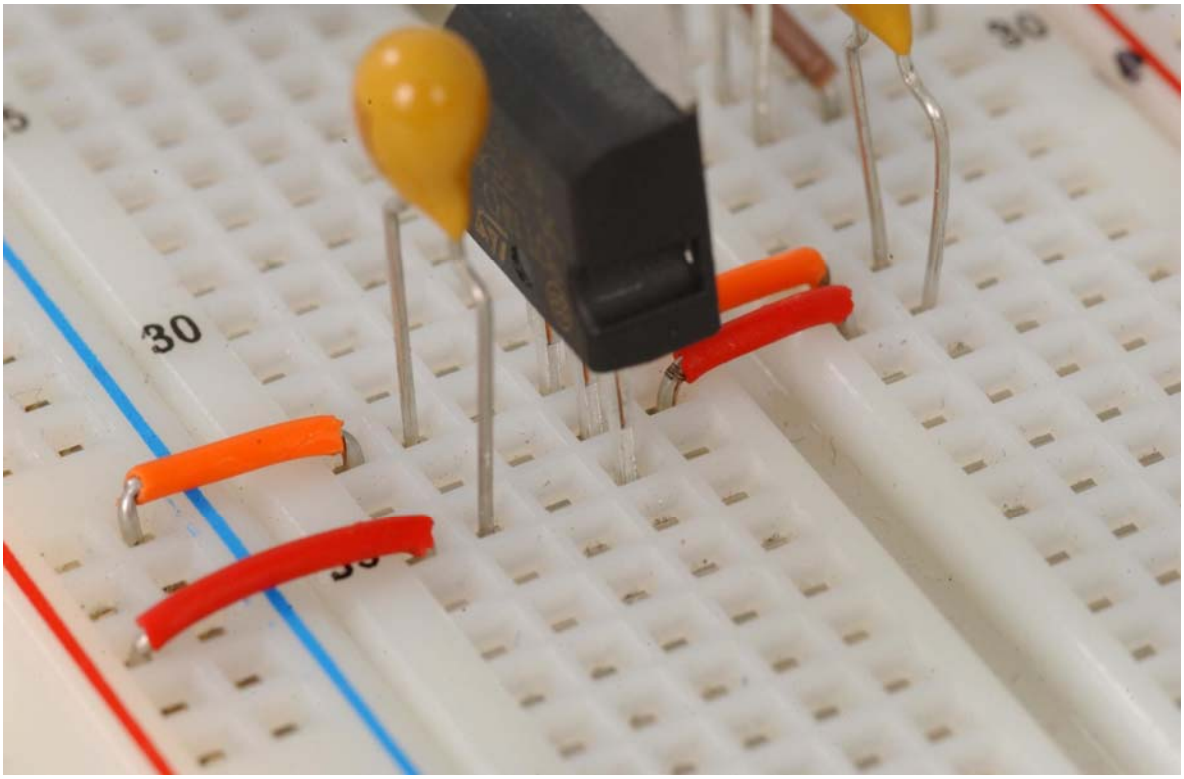
Arduino function					Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	22	GND	GND
GND	GND	8	21	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)	digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

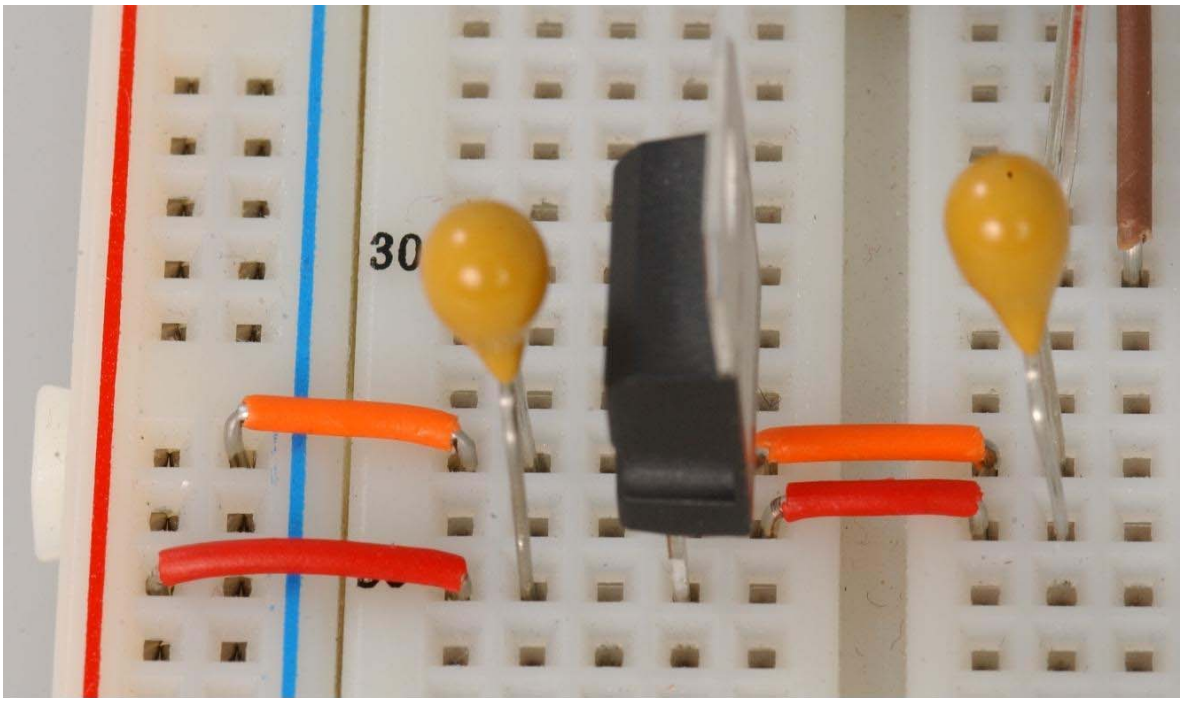
Digital Pins 11,12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17,18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.



The remaining components are the header (for setting up a USB-to-Serial programmer) and the components to create a 3.3V power circuit. You may need 3.3 volts for various sensors or other ICs, so we will add those components next. Find some space near the bottom of the breadboard to place the 3.3V regulator (LM1117T-3.3). This is also a TO-220 package, but the pin out is different from the 7805T 5V regulator. With the chip facing you and the leads pointed down, pin 1 (left leg) is ground. Pin 2 is the output voltage side that will produce 3.3 volts. Pin 3 (right leg) is the input power side. Place a jumper from the ground rail to pin 1. Add another jumper from the 5V power rail to pin 3 of the voltage regulator. I used a couple small pieces of jumper wire to bring the output side to its own row of pins on the breadboard. Install one 10µF tantalum capacitors between the power and ground pins on the input side, and install the other 10uF capacitor between the power and ground of the output side. **Note:** The tantalum capacitors are polarized, so be sure to install them correctly. The printed face should have a (+) sign on it, but if it doesn't, the longer leg is the positive side. See images below.

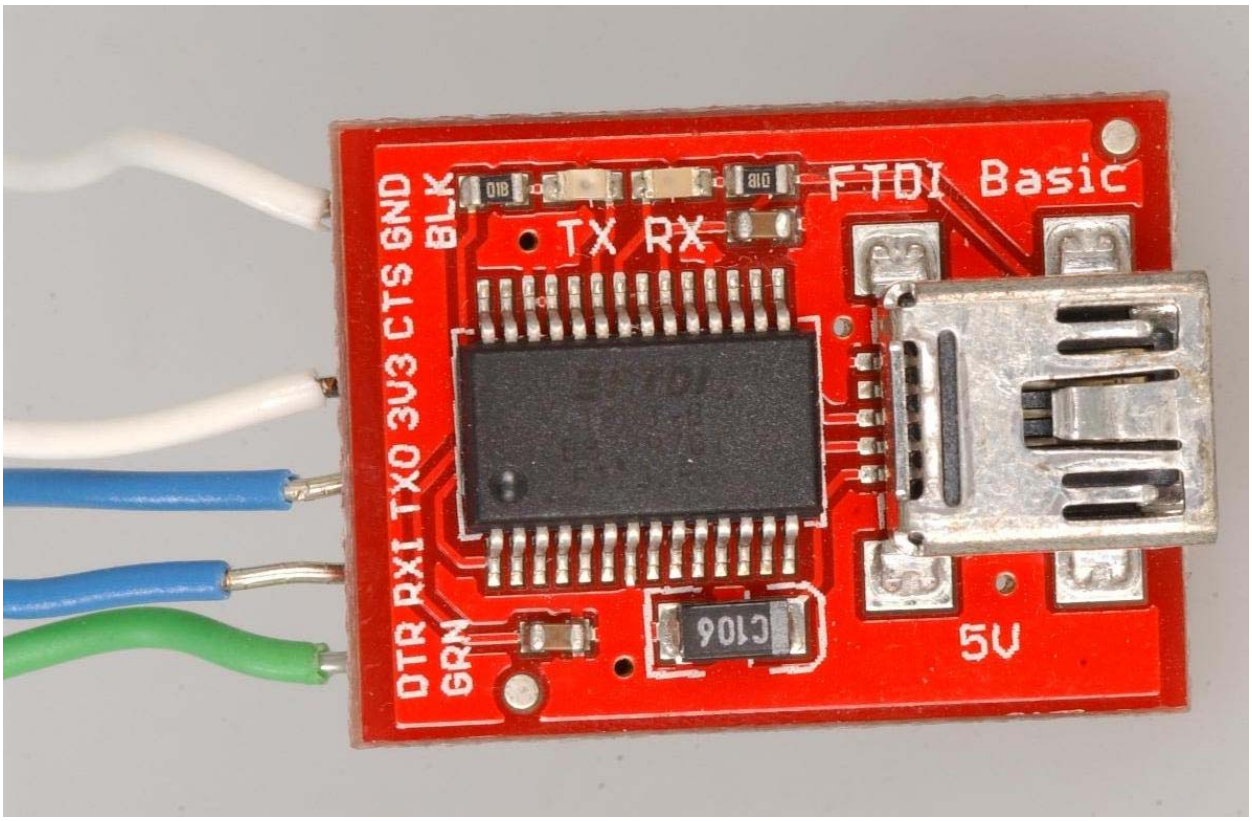


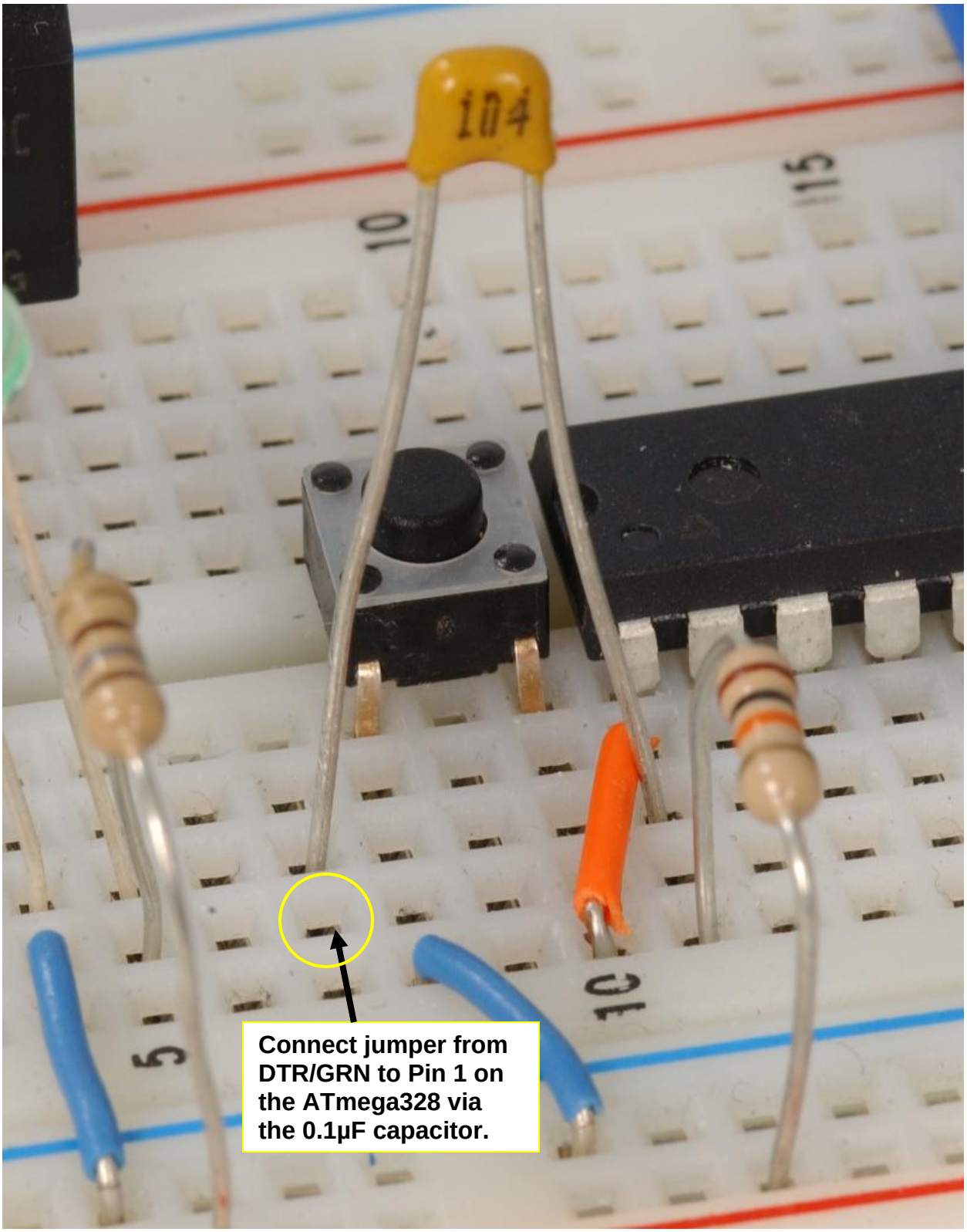


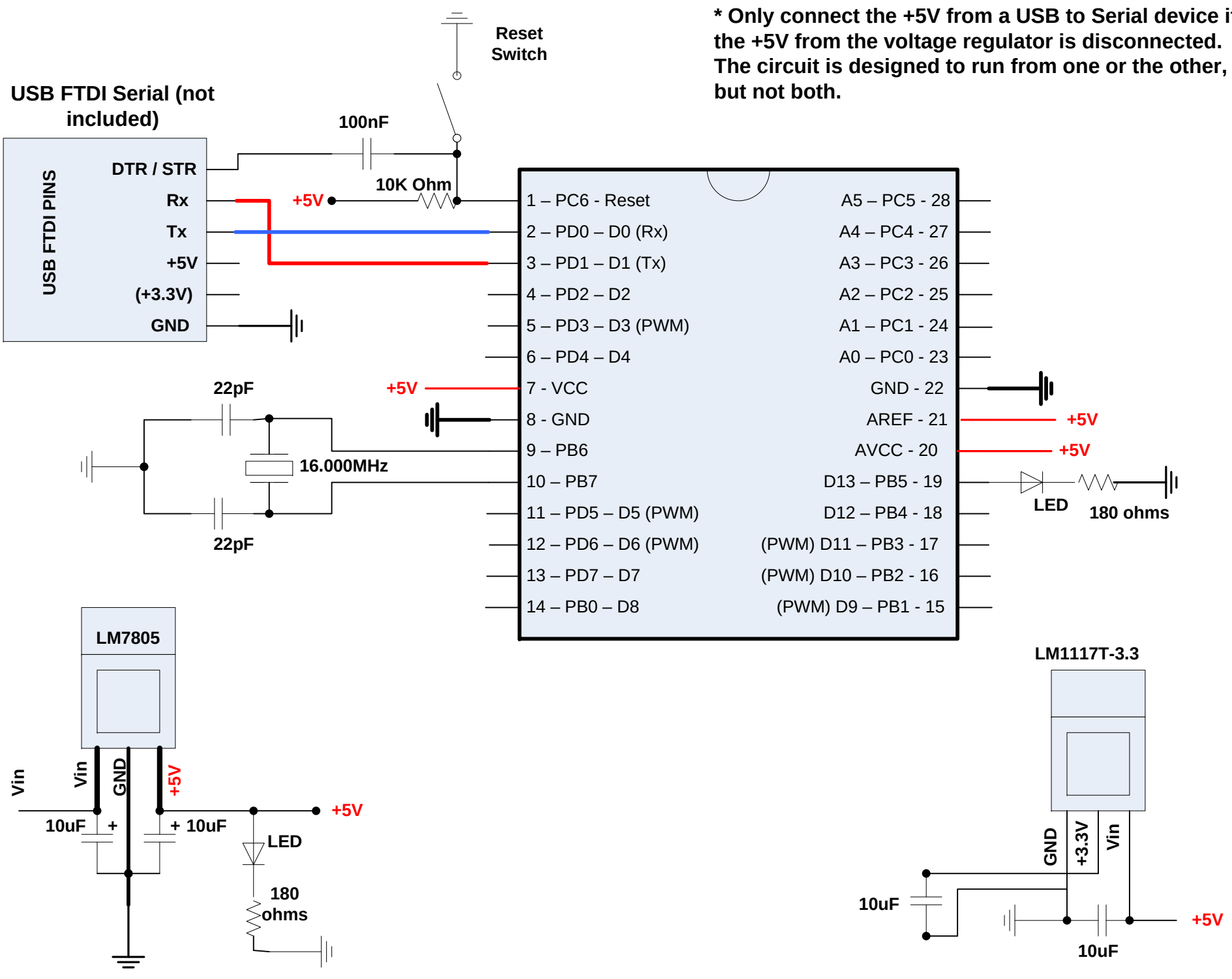


If your ATmega328 chip is preprogrammed, you should be in business. If not, there are a few more steps necessary to program it. You will need a USB-to-Serial device. In my example, I am using the FDTI Basic Breakout Board (5V) (P/N: 2117341). If you just want to get it working, you can skip installing the 6-pin header (P/N: 153700) and just run jumper wires straight from the USB-TTL header to the appropriate pins on the breadboard. When you do install the 6-pin header, make sure the pins are routed correctly for the serial device you choose. The pins on the breakout board are labeled with 3-digit names. According to most tutorials, you only need four pins; RXI, TXO, 3V3 (even though it is 5V unless you break the jumper on the board), and GND. Setting it up this way left me scratching my head because it wouldn't upload my sketch. After doing a little research, I discovered the microcontroller needs a perfectly timed press of the reset button to ready the chip to be programmed. Nowhere could I find what that magic timing was, but I did find out the breakout board has a pin called DTR/GRN which sends a signal to the reset pin when hooked up properly. And by "properly", I mean connect a jumper wire from (DTR/GRN) on the breakout board to Pin 1 of the ATmega328 via a 0.1µF ceramic capacitor. Bingo!

Pin on breakout board	**Pin on microcontroller**
DTR/GRN	Pin 1 (RESET) via 0.1µF cap
RXI	Pin 3 (TX) (digital pin 1)
TXO	Pin 2 (RX) (digital pin 0)
3V3	5V power source
CTS	(unused)
GND	Ground


DTR RXI TXO 3V3 CTS GND
BLK
GRN
TX RX
FTDI Basic
C106
5V


104
10
15
5
10
Connect jumper from DTR/GRN to Pin 1 on the ATmega328 via the 0.1µF capacitor.


* Only connect the +5V from a USB to Serial device if the +5V from the voltage regulator is disconnected. The circuit is designed to run from one or the other, but not both.
Reset Switch
USB FTDI Serial (not included)
USB FTDI PINS
DTR / STR
Rx
Tx
+5V
(+3.3V)
GND
100nF
10K Ohm
+5V
22pF
16.000MHz
22pF
+5V
1 – PC6 - Reset
2 – PD0 – D0 (Rx)
3 – PD1 – D1 (Tx)
4 – PD2 – D2
5 – PD3 – D3 (PWM)
6 – PD4 – D4
7 - VCC
8 - GND
9 – PB6
10 – PB7
11 – PD5 – D5 (PWM)
12 – PD6 – D6 (PWM)
13 – PD7 – D7
14 – PB0 – D8
A5 – PC5 - 28
A4 – PC4 - 27
A3 – PC3 - 26
A2 – PC2 - 25
A1 – PC1 - 24
A0 – PC0 - 23
GND - 22
AREF - 21
AVCC - 20
D13 – PB5 - 19
D12 – PB4 - 18
(PWM) D11 – PB3 - 17
(PWM) D10 – PB2 - 16
(PWM) D9 – PB1 - 15
+5V
+5V
LED
180 ohms
LM7805
Vin
Vin
GND
+5V
10uF
+
+ 10uF
+5V
LED
180 ohms
LM1117T-3.3
GND
+3.3V
Vin
10uF
+5V
10uF