

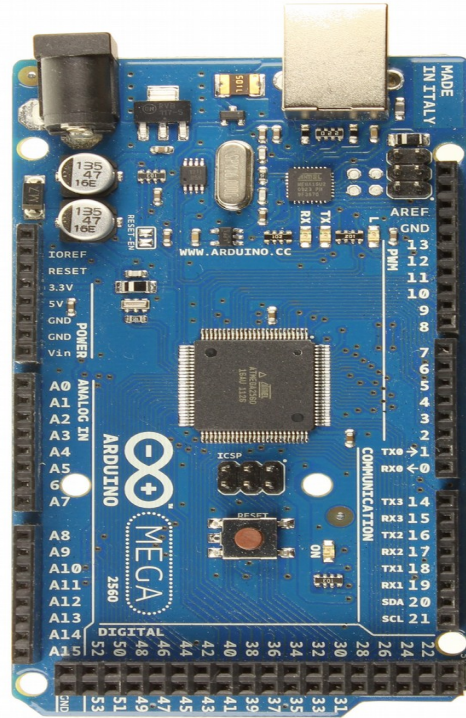


# The Seminar in a Nutshell

## Station Automation == Arduino

Inputs:

GPIO Pins  
Analog Pins  
Serial  
Ethernet

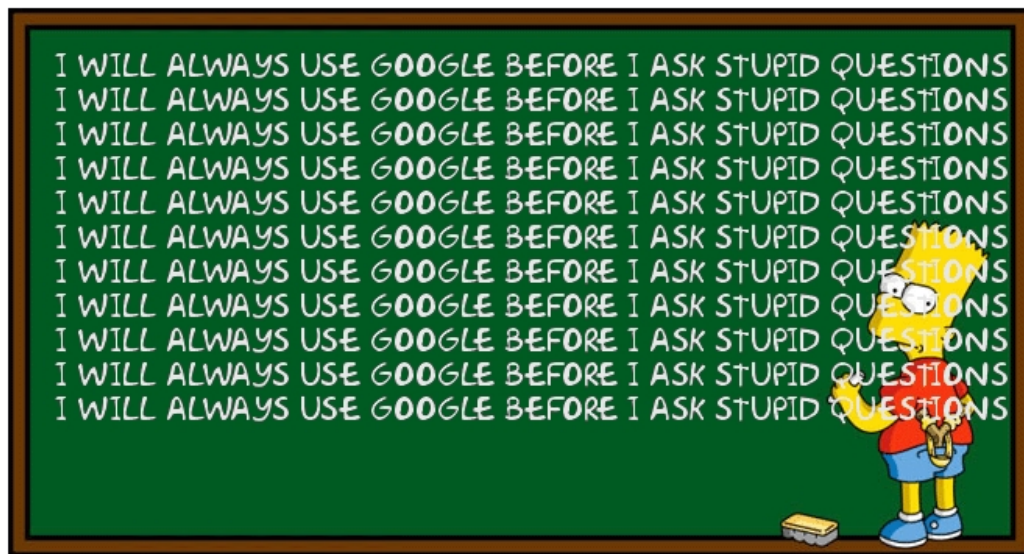


Outputs:

GPIO Pins --> Relays  
Analog Pins --> PWM  
                          motor  
                          control  
Serial --> USB-Serial  
Ethernet --> Browser  
                  Database

# Google™ Is Your Best Friend ...

... but it seems you haven't recognized it yet!



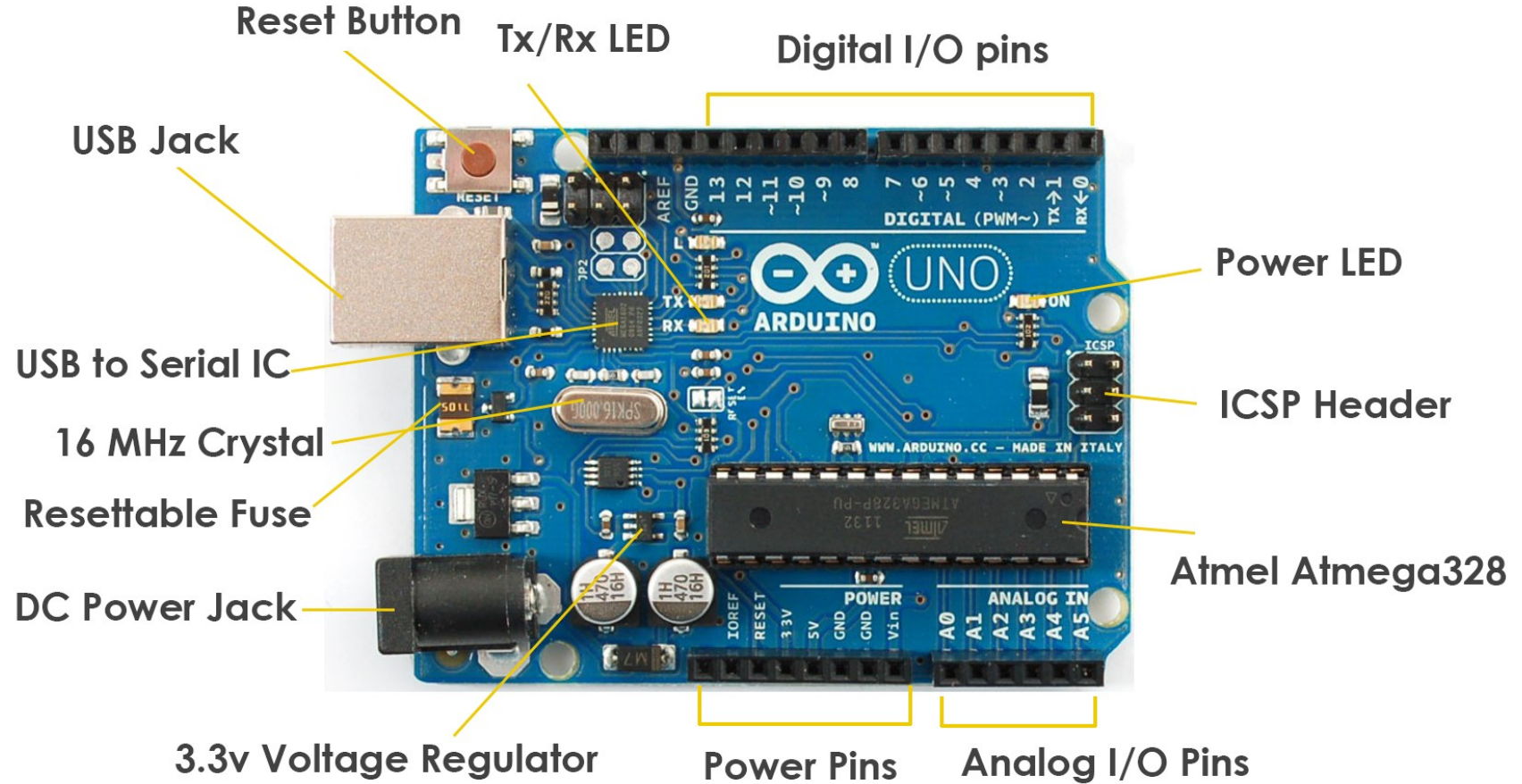
Google™

Google Search

Google is your best Friend!!

Somebody thinks you are bothering people with your stupid questions, because you are too lazy to search on your own. This Person sent this link to YOU to point out that fact.

I hope you use your best friend Google next time :-)



# Station Automation

- In the past, a typical station had only manual control of:
  - Frequency - spin the knob
  - PTT Switching between radios (or multiple switches)
  - Mic switching between radios (or used multiple mics)
  - Receive audio switching between radios (or used multiple headphones/speakers)
  - CW key/paddle switching between radios (or used multiple keys/paddles)
  - Antenna azimuth / elevation control
  - RF output power, VSWR monitoring, etc.
- Each of these functions can be automated

# Station Automation

- Why add automation?
  - Because having to switch focus between the logging computer and mechanical switches / or – having to remember to set mechanical switches:
    - Decreases efficiency, potentially decreasing number of QSOs
    - Increases likelihood of errors that reduce number of QSOs or damage equipment (or both)
    - Increases operator fatigue
  - And adding automation solves these problems
  - Because bringing data display to the computer screen improves ergonomics and increases the likelihood that data will be seen and acted on appropriately
  - Because automation is necessary for remote operation, where mechanical switches and data displays at the remote location cannot be accessed from operator location
  - Because it is a fun technical challenge likely to be successfully implemented

# Why Arduino?

- Single Board Microcontroller
- Open Source
- Cheap
- Largest User Community by Far
  - Lots of documentation on the web
  - Plenty of already-written code on the web

# Why Arduino?

- Simple to Learn and Use
- Excellent IDE (Integrated Development Environment)
- Powerful enough for most projects
- Relatively rugged



# What Can You Do With an Arduino?

- Switching of devices
  - Simple Power On/Off
  - Switching a function on and off (e.g.PTT)
  - Switching a device among several radios
    - MIC, CW Key/Keyer, PTT
  - Switching receive audio going to headphones from among several radios
  - Bandswitching transverters, amplifiers, antennas
  - Switching among several cameras
  - Switching among several antennas on a given band

# What Can You Do With an Arduino?

- Monitoring of devices
  - RF Power out / Reflected Power / SWR
  - Voltage / current
  - Device Status
    - On/Off
    - Show which of several states is selected / active

# What Can You Do With an Arduino?

- Controlling and Monitoring Devices
  - Antenna Azimuth / Elevation Control: K3NG Rotator Controller
- Special Functions
  - CW Keyer
  - Fan Speed Controller
  - Digital VFO Control (e.g.DDS-60)
  - APRS Tracker
  - WSPR Beacon
  - Antenna Tuner Controller
  - Packet Radio Controller

# What Can You Do With an Arduino?

- Beacon Controller
- Code Practice Generator
- CW Decoder
- Waveform Generator
- SDRs - CW and JT65 transceivers
- LF transmitter
- Battery Charge Monitor
- Station Power Monitor
- Voice Memory Keyer

# What can you do with an Arduino?

## "mini-Maestro" for HP/PowerSDR



# What can you do with an Arduino?

## "mini-Maestro" for HP/PowerSDR

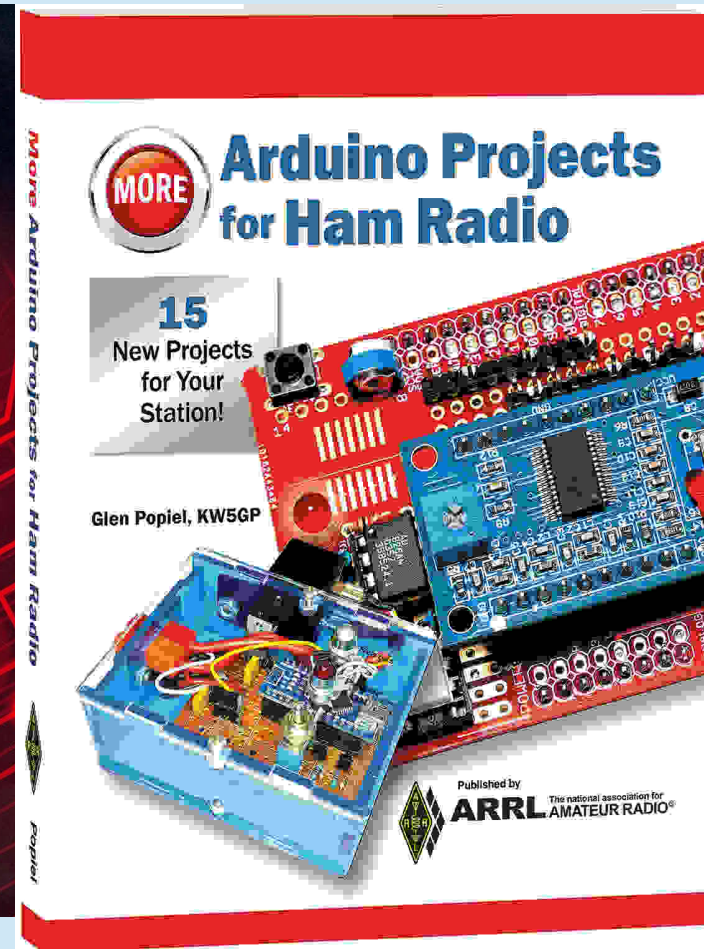
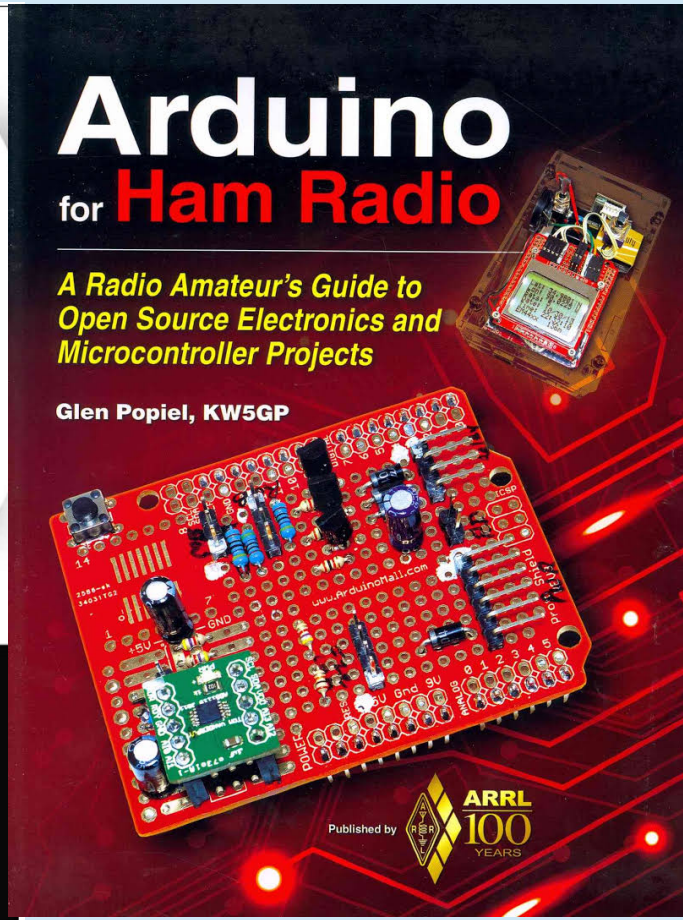


# What can you do with an Arduino?

”mini-Maestro” for HPSSDR/PowerSSDR

- [http://www.nicklebyhouse.co.uk/index.php/software-define-d-radio/237-a-start-in\\_sdr\\_part\\_13\\_the\\_odin\\_console](http://www.nicklebyhouse.co.uk/index.php/software-define-d-radio/237-a-start-in_sdr_part_13_the_odin_console)
- G8JJJ and LA2NI
- Just announced this past Tuesday, 4/17/18
- Uses Arduino Due:
  - 32 bit ARM core
  - 512KB Flash, 96KB SRAM
  - 3.3V bus

# What Can You Do With an Arduino?





# What Arduinos are Available?

- Genuino vs Knock-Offs from China

- Knock-Offs are much cheaper

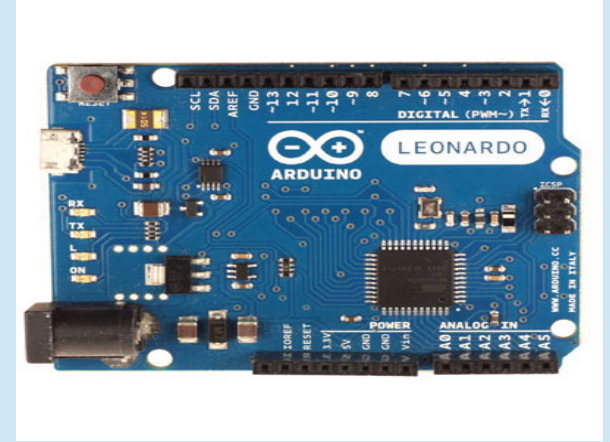
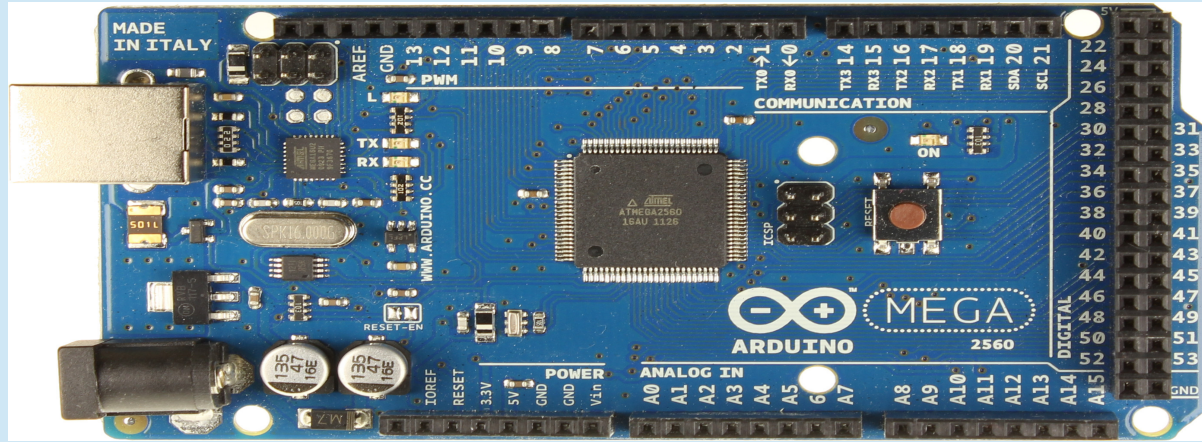
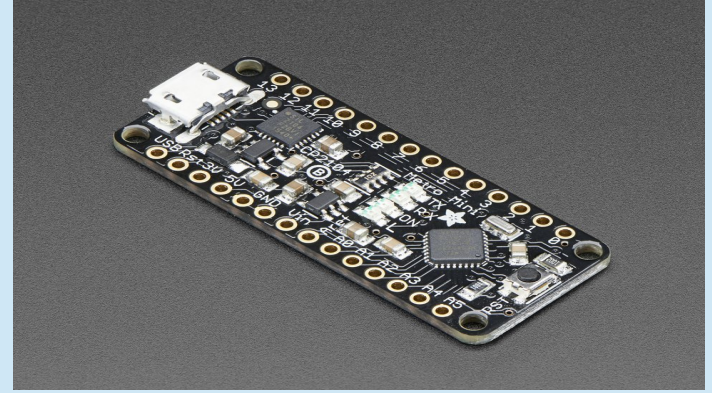
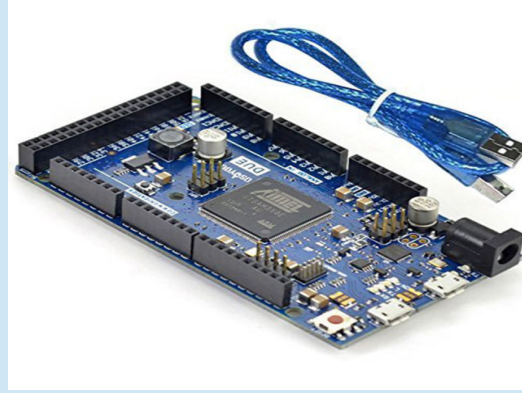
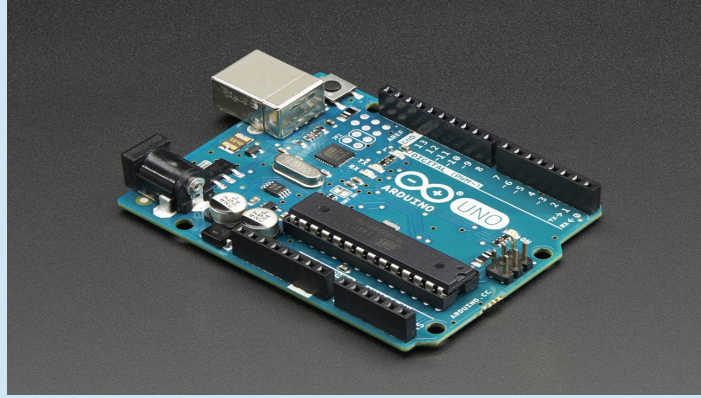
Model	Genuine	Knock-Off
Uno Rev3	\$22.00	\$4.62
Mega 2560 R3	\$38.50	\$10.58

- Knock-Offs may require driver download
- Knock-Offs have worked very well
- Knock-Offs are DEFINITELY Cost Efficient

# What Models of Arduino are Available?

- 10 “retired” Arduino models
- 17 active models:
  - Entry Level Boards: **Uno**, Leonardo, 101, Esplora
    - Modules: Micro, Nano, Mini
  - Enhanced Boards: **Mega**, Zero, Due, Mega ADK, M0, M0 Pro
    - Modules: Mkr Zero
  - Internet of Things Boards: Yun, Ethernet, Tian, Industrial 101, Leonardo Eth, Mkr Fox 1200, Mkr Wan 1300, Mkr GSM 1400
    - Modules: Mkr1000, Yun Mini
  - Boards physically larger than modules, have common form factor, accept shields, more expensive than modules

# Arduinos



MCU – 8 bit unless noted	GPIO	Analog I/O	LPT	U S B	Enet	Max ma sourced	Flash/ SRAM/ EEPROM KB	GPIO V	\$\$\$\$
<b>Arduino UNO</b>	20	6/0	Y	Y	N	20	32/2/1	5	5.13
Arduino Nano	22	8/0	Y	Y	N	40	32/2/1	5	2.93
Arduino Leonardo	20	12/0	Y	Y	N	40	32/2.5/1	5	10.93
Arduino Metro Mini 328	20	6	Y	Y	N	20 (40)	32/2/1	5 (3.3)	17.94
<b>Arduino Mega 2560 R3</b>	54	16/0	Y	Y	N	40	256/8/4	5	10.43
Arduino Due 32bit	54	12/2	Y	Y	N	3 or 15	512/96/N	3.3	15.10
Arduino Zero 32bit	20	6/1	Y	Y	N	7	256/32/N	3.3	15.00



GOOD



BAD

# Memory Types

- FLASH – Program space where program is stored
  - Non-volatile
- Static Random Access Memory (SRAM) – where variables are written to; also called “dynamic memory”
  - Volatile
- EEPROM – memory space for storing long-term information
  - Non-volatile

# What, No Internet? Use an Ethernet Shield!

Arduino Ethernet Shield  
Ebay \$5.78 UNO-type  
\$6.40 NANO-type



# Are there Other Shields?

## Arduino Shield List

Pin usage details for 317 shields from 125 makers, and counting!

Search:  GO

> [Home](#)

- [4D Systems](#) (4)
- [Adafruit Industries](#) (9)
- [AeroQuad](#) (2)
- [Andre Concalves](#) (2)
- [antrax Datentechnik](#) (1)
- [Applied Platonics](#) (1)
- [ArduCapSense](#) (1)
- [Arduino](#) (3)
- [Argent Data Systems](#) (1)
- [Asynclabs](#) (3)
- [Batsocks](#) (2)
- [Ben Combee](#) (1)
- [Bhasha Technologies](#) (2)
- [Bliptronics](#) (2)
- [Blushing Boy](#) (1)
- [Carlos Neves](#) (1)
- [Chesters Garage](#) (1)
- [Chips To Bits](#) (1)
- [Circuit Ideas Design](#) (1)
- [Circuits At Home](#) (2)
- [CISECO](#) (4)
- [Collin Schulz](#) (1)
- [Conductive Resistance](#) (1)
- [Control Connection](#) (1)
- [Creatron Inc](#) (1)
- [Critical Velocity](#) (5)
- [Crittter and Guifari](#) (1)
- [Curious Inventor](#) (2)
- [CuteDigi](#) (8)
- [Damien Good](#) (1)
- [Dexter Industries](#) (1)
- [DFRobot](#) (16)
- [Dr Michael Kroll](#) (1)
- [Dreaming Robots](#) (1)
- [DroidBuilder](#) (1)
- [DSS Circuits](#) (1)
- [Dynamic Perception](#) (1)
- [Emartee](#) (1)
- [Eric Rogers](#) (1)
- [EtherMania](#) (1)
- [Evil Mad Science](#) (2)
- [Excamera Labs](#) (1)
- [Faz Jaxton](#) (1)
- [FlamingoEDA](#) (1)
- [Freelonics](#) (12)
- [Futura Elettronica](#) (2)
- [Galileo Z](#) (4)
- [GeekOnFire](#) (1)
- [GfxHax](#) (1)
- [GinSing](#) (1)
- [Gravitech](#) (1)
- [Homerosters](#) (1)
- [HW Kitchen](#) (1)
- [Tead Studio](#) (6)
- [Jee Labs](#) (1)
- [Jimmie Rodgers](#) (1)
- [John Liu](#) (3)
- [Knutset](#) (1)
- [Lars Schumann](#) (3)
- [Libellium](#) (6)
- [LinkSprite](#) (5)
- [Liquidware](#) (12)
- [Logos Electromechanical](#) (1)
- [Low Voltage Labs](#) (1)
- [Luke Weston](#) (1)
- [macetech](#) (3)
- [Maker Shed](#) (1)
- [Mark Sproul](#) (1)
- [Max Pierson](#) (1)
- [Mayhew Labs](#) (2)
- [MCI Electronics](#) (4)
- [McLaughlin Engineering](#) (1)
- [MightyOhm](#) (2)
- [Mitek](#) (1)
- [Modern Device](#) (1)
- [Multilogica](#) (1)
- [Narbotic Instruments](#) (1)
- [Neuroelec](#) (2)
- [NKC Electronics](#) (4)
- [Nootropic Design](#) (3)
- [North And Nash](#) (1)
- [Nu Electronics](#) (9)
- [Ocean Controls](#) (1)
- [Open Electronics](#) (1)
- [PDK Solutions](#) (1)
- [Photoduino](#) (1)
- [Pololu](#) (1)
- [Practical Maker](#) (5)
- [Protoino](#) (2)
- [Ray's Hobby](#) (1)
- [Renbotics](#) (2)
- [RepRap Research Foundation](#) (1)
- [Ro-Bot-X Designs](#) (1)
- [Robot Power](#) (1)
- [RobotPirate](#) (1)
- [Rocket Scream](#) (1)
- [Rogue Robotics](#) (1)
- [Rugged Circuits](#) (6)
- [Samurai Circuits](#) (3)
- [Scattered Mind](#) (1)
- [Schmelle2](#) (6)
- [Seeed Studio](#) (14)
- [Shieldstudio](#) (2)
- [SK Pang Electronics](#) (3)
- [Small Room Labs](#) (1)
- [Smart Energy Groups](#) (1)
- [Snootlab](#) (6)
- [Solarbotics](#) (2)
- [SonikTech](#) (1)
- [Sparkfun](#) (26)
- [Spikenzielabs](#) (4)
- [Sunhayato](#) (2)
- [SWFLTEK](#) (1)
- [Synthetos](#) (1)
- [Unified Microsystems](#) (1)
- [Unsped](#) (1)
- [Watterott](#) (3)
- [Wavical Technologies](#) (1)
- [Wayne and Layne](#) (1)
- [Wicked Device](#) (3)
- [Wingshield Industries](#) (1)
- [Wise Time](#) (2)
- [Yawp](#) (2)
- [Zach Hoeken](#) (1)

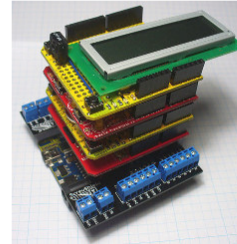
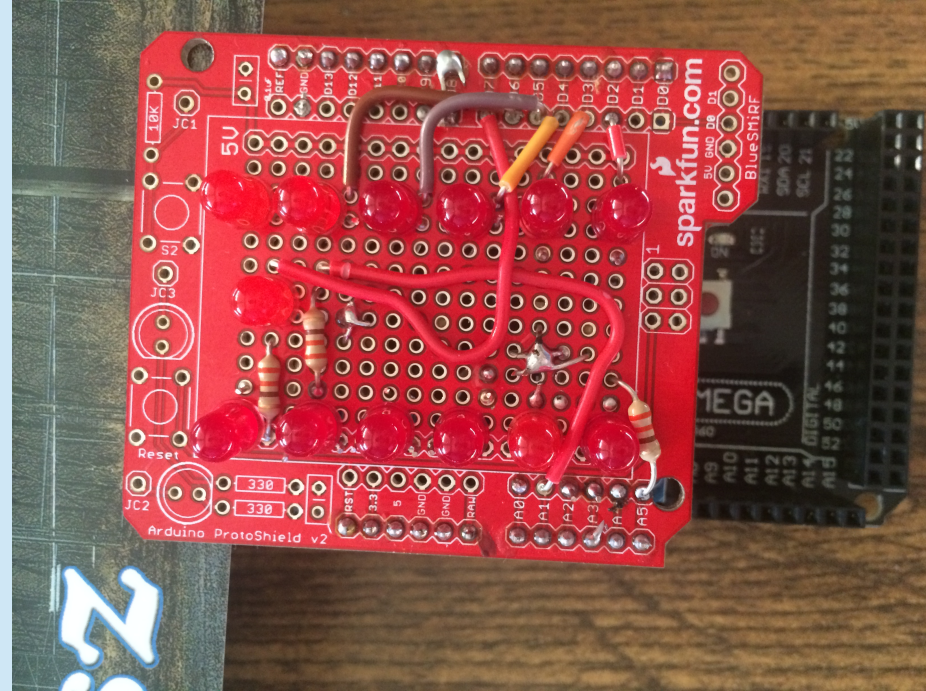
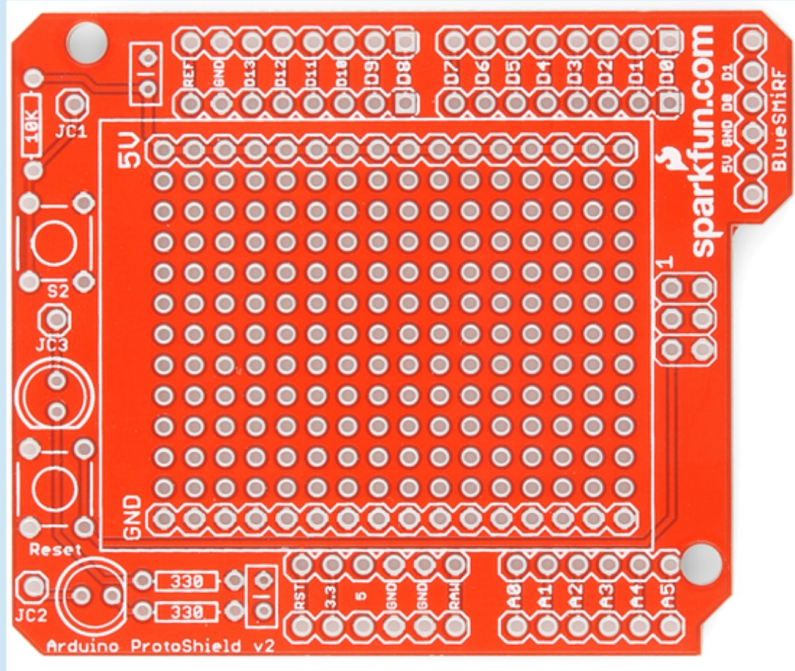


Photo: [John Boxall](#)

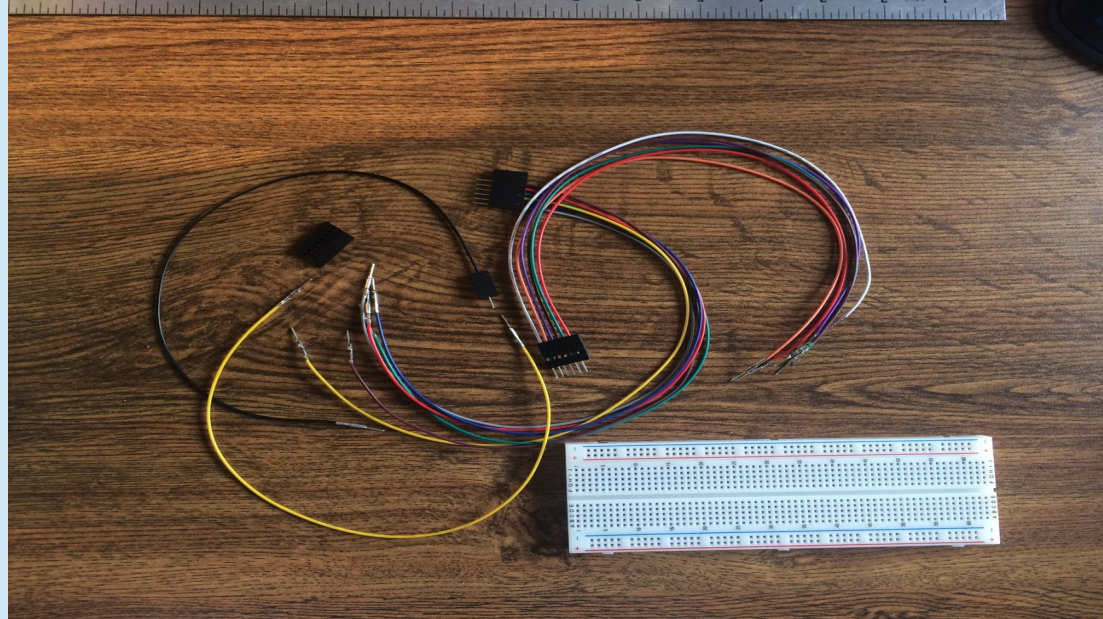
# Prototyping Shields



Arduino ProtoShield – Bare PCB \$4.95 from Sparkfun  
<https://www.sparkfun.com/products/11665>



# Prototyping Before You Make a Shield / Cape / Hat

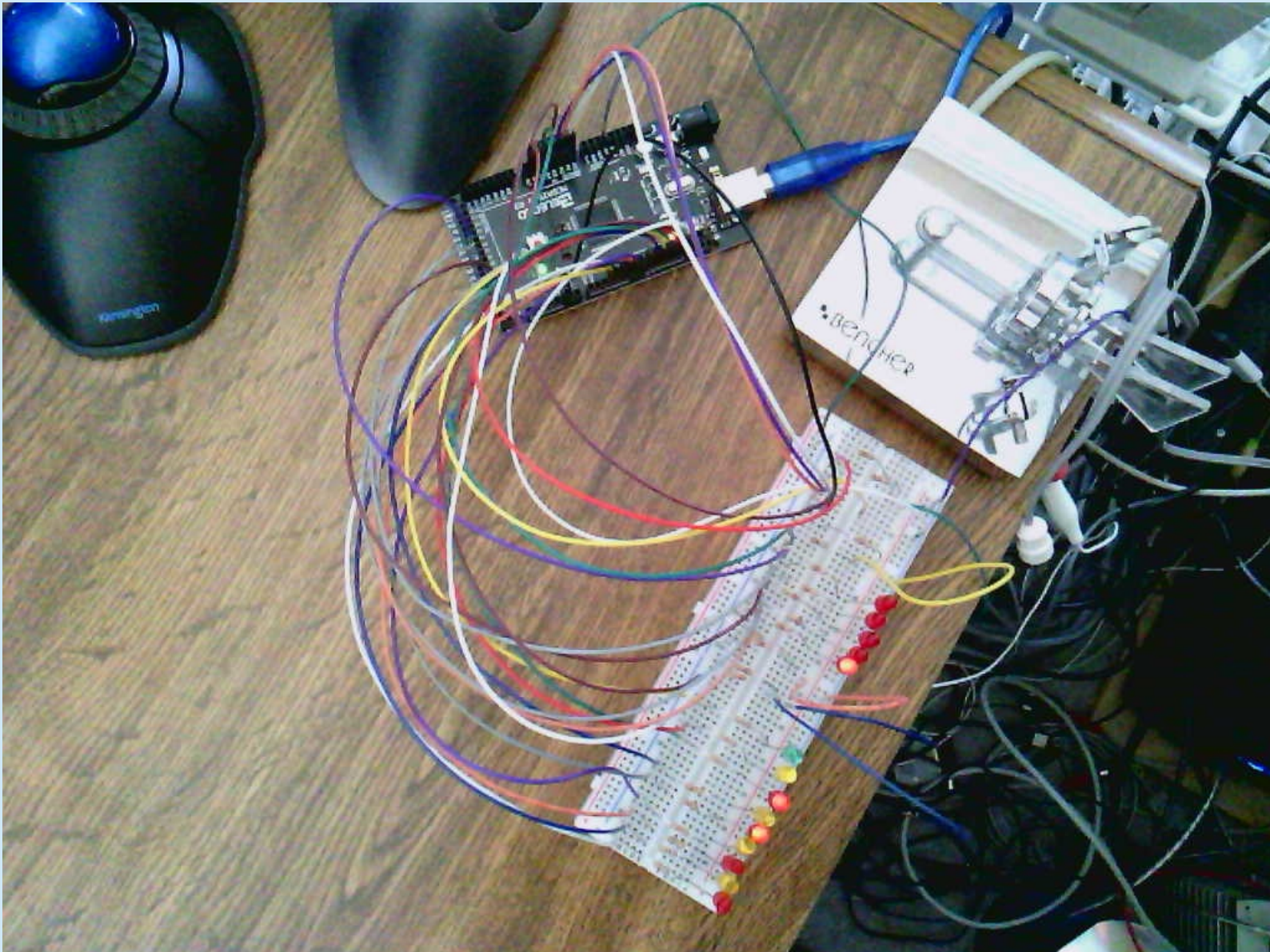


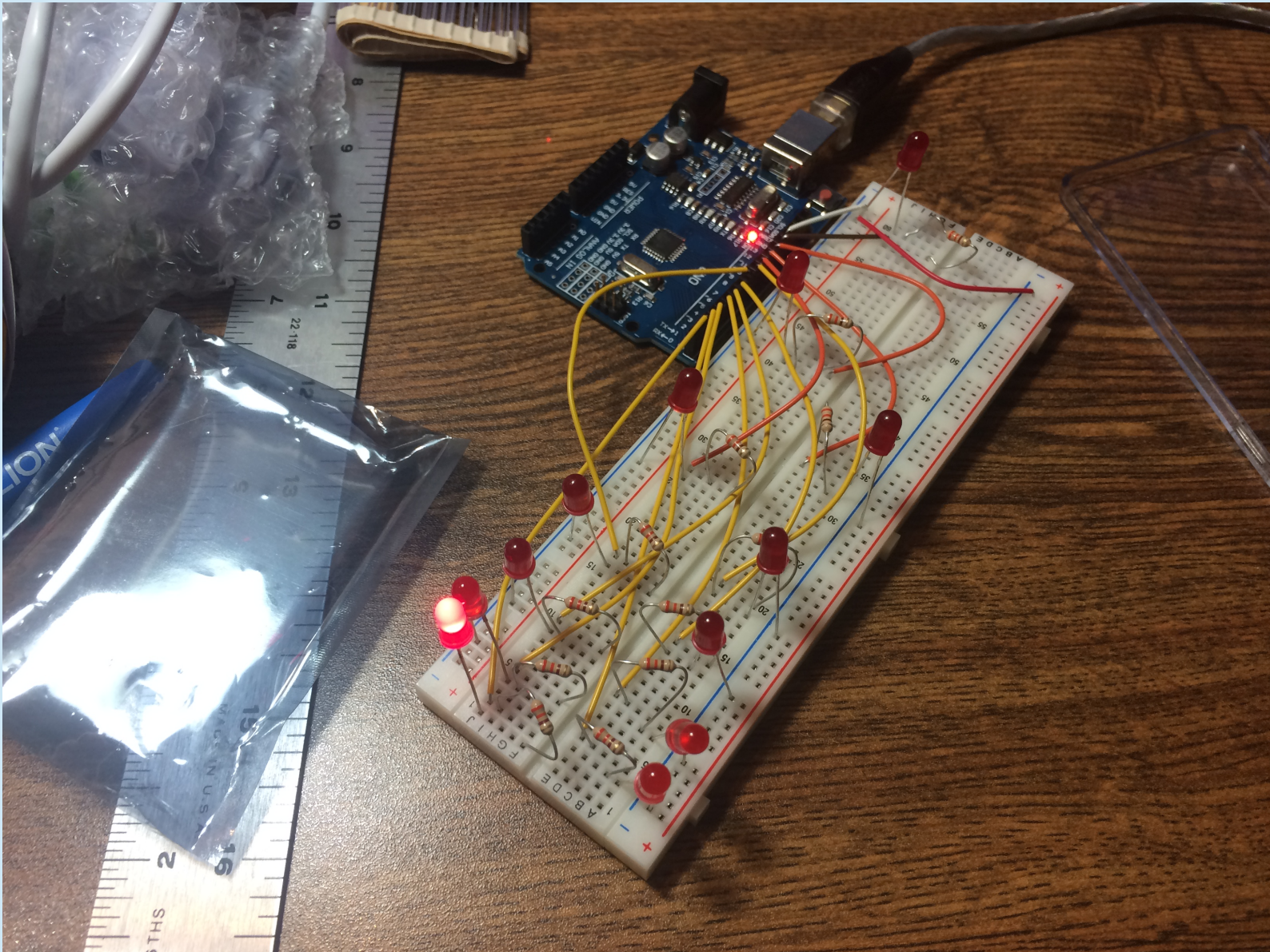
Wires with Pre-Crimped Terminals -and- Crimp Connector Housings

<https://www.pololu.com/category/39/cables-and-wire>

MB-102 Breadboard Solderless Protoboard PCB Test Board 830 Points

<https://www.ebay.com> \$0.99





REMEMBER: WITH GREAT  
POWER COMES GREAT  
CURRENT SQUARED  
TIMES RESISTANCE.



OHM NEVER FORGOT HIS  
DYING UNCLE'S ADVICE.