Hardware Hacking for us dumb software guys

ARDUINOS

My Story

- Done software since 1999
  - Degree in Computer Science
- Done a little hardware over the years
- About five years ago bought a Microchip PIC Kit
  - The labs were cool and easy to get working
  - Anything practical was difficult (low level assembly)

Defcon 17

- Bought all the hardware hacking villiage kits at Defcon 17
- Picked up an Arduino shortly after Defcon
- Was given a decent soldering iron
- Started building my own projects

Hardware hacking is daunting for many computer hackers

- understanding all the electrical components and complex circuits
- cost
- steep learning curve for some languages used in electronics (i.e. assembly)
- lack of good information and good community support for many devices

Microcontrollers mix hardware and software

- You can drop code on them instead of building complex circuits in some places
- You can reprogram them to be something else whenever you want

Arduinos

- Open-Source hardware platform
- All-in-one prototyping board
- Easy to learn language
- Free, Open Source IDE
- Strong Community Backing
- Cheap (around $30)

Kind of like the hardware equivalent of a scripting language

What's an Arduino?

Arduino Duemilanove

- ATmega328p
- 14 Digital I/O pins
  - 6 are PWM Pins
- 6 Analog pins
- 32 KB Flash
- 16 MHz

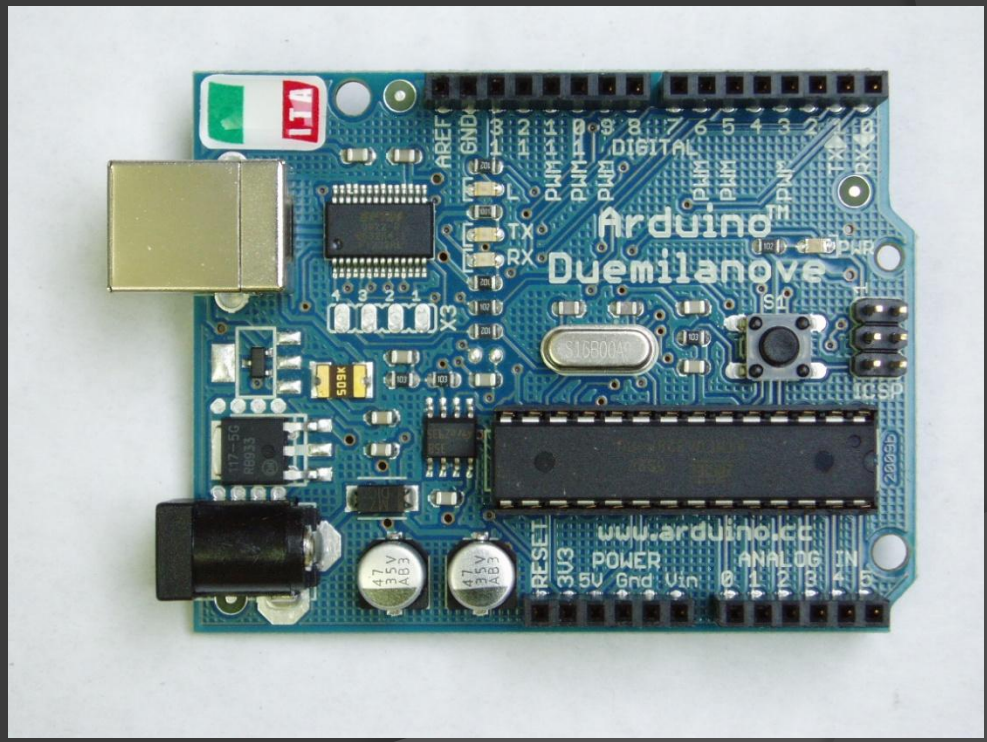


*Image source – Wikipedia.org Arduino

What else is an Arduino?

- Since it's open source hardware there are many other versions of Arduino compatible boards
- Some boards are only IDE compatible, others are also pin compatible

Other Arduinos

- Arduino Mega
- Sparkfun's Arduino Pro and Mini
- Evil Mad Scientist Diavolino
- Fio
- Teensy
- Freeduino
- Seeeduino
- Lilypad
- Ardweeny
- Bare bones

Arduino IDE

- Syntax Highlighting
- Compiles Code
- Uploads to the Arduino
- Serial console

Arduino Speak

- Shield – componets that fit on top of an Arduino
- Sketch – Arduino code
- .pde – Arduino code file type

Arduino Language

- Like C, but simpler
- Has methods for common tasks such as reading and writing to pins, serial, etc.
- Relatively easy to include external libraries

Sketch

- Define global variables and/or constants
- A setup function for things that must be run when the program first starts up
- A loop function the Arudino will loop through to run the program
- Any other user defined functions

Blink LED

```
int ledPin =  13;    // LED connected to digital pin 13

void setup()   {
   pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH);   // set the LED on
  delay(1000);                  // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);                  // wait for a second
}
```

Google Trends

*Source Google Trends

Arduinos don't work for every project

- While they're pretty cheap they're expensive when you want to make a lot of devices
- Not incredibly powerful
- No parallel computing (have interrupts)

Using Available Shields

- Many shields are available, including:
  - Ethernet Shield
  - Xbee Shield
  - Motor Shields
  - Wave Shield
  - Nixie Tube Shield
  - LCD Shield
  - Cellular Shield
  - . . .
- Advantage – no tools necessary

Using Shields

Xbee Shield

- 2.4 Ghz
- 250 kbps max data rate
- 128 bit encryption
- From a few hundred feet to several miles in range

Xbees and Security

- Can be used for out of band communication
- Even though they run on 2.4 GHz it's unlikely they'd be detected during a war driving assessment (not 802.11)
- Can reprogram your Arduino remotely using an Xbee

Ethernet Shield

- Can function as a client or server
- Up to 4 simultaneous connections

Xbee Shield and Ethernet Shield Together

- Place an Arduino on a network with an Xbee Shield and an Ethernet Shield to launch attacks on the network
- Xbee traffic will probably never be detected
- The device is small and may go unnoticed for long periods of time

Making your own Shields

- You can use Arduinos without doing shields, but it's a good convention to follow
- It's easy to make your own shields
- You can use a proto shield, custom PCB or multipurpose PCB

- You'll need a soldering iron and possibly other tools

Serial

- Easy to use
- Arduino digital I/O pins 0 and 1 are hardware serial pins
- Other digital I/O pins can be software serial pins
- Example devices:
  - Parallax RFID Reader
  - GPS Units
  - LCDs
  - Various Integrated Circuits

I2C

- Easy to use
- Analog pins 4 and 5 are I2C pins
- I2C is a bus
  - Can connect up to 128 devices
  - Each device has it's own address
- Examples:
  - Centipede Shield
  - Wii Nunchuck
  - LCDs

Reverse Engineering Hardware using I2C and Serial

- Arduino can be connected to other devices that support I2C and/or Serial to try and determine how they work.
- Examples:
  - Connecting to serial pin on a Segate hard drive
  - Connect to I2C bus of a Wii

GPS Tracking Device

- Uses a serial GPS unit
- Can log data to an SD card or broadcast it (i.e. Xbee or cellular).
- Used to track down stolen good
- Used to see where people are going

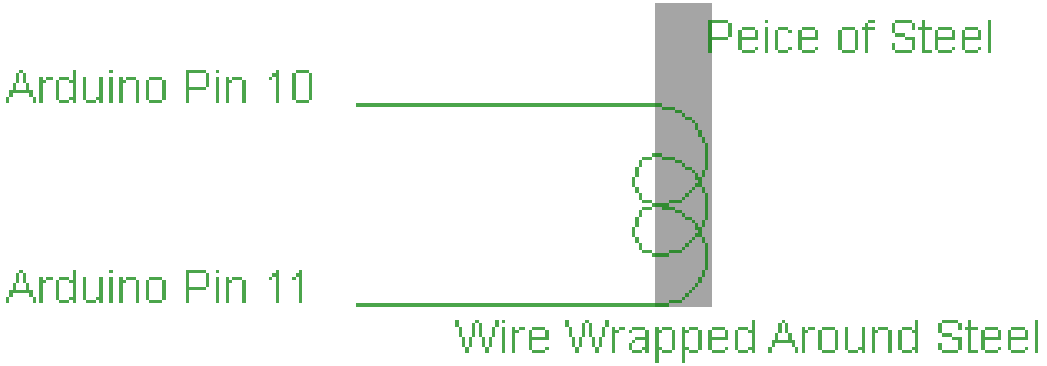
Magnetic card reader and spoofer

- by connecting a magnetic stripe card reader to an Arduino you can read the data from the magnetic stripe card
- using an electromagnet emulate a card swiping on a card reader
- this device could be used to read a magnetic stripe card, then spoof the data on a magnetic card

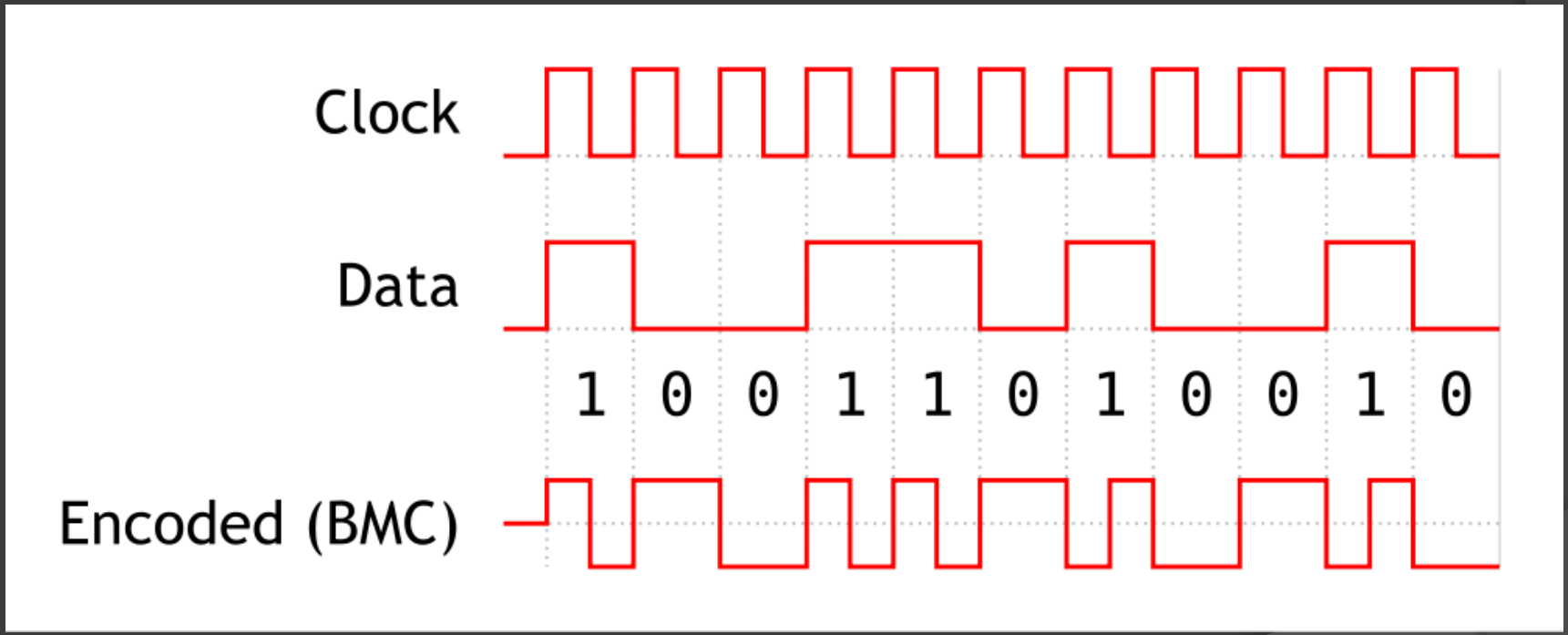
Simply Put

- Create an electromagnet
- Switch the polarity back and forth to make the reader think a card is being swiped

Schematic


Peice of Steel
Arduino Pin 10
Arduino Pin 11
Wire Wrapped Around Steel

Biphase mark code



*Source Wikipedia.org Biphase mark code

Code

```
if(lowOrHigh == 1)
  {
    if(clockHalf == 0)
    {
      digitalWrite(rightPin, LOW);
      digitalWrite(leftPin, HIGH);
      delayMicroseconds(clockSpeed);
      digitalWrite(leftPin, LOW);
      digitalWrite(rightPin, HIGH);
      delayMicroseconds(clockSpeed);
      //clockHalf = 1;
    }
```

More Code

```
if(clockHalf == 0)
{
  digitalWrite(rightPin, LOW);
  digitalWrite(leftPin, HIGH);
  delayMicroseconds(clockSpeed * 2);
  clockHalf = 1;
}
```

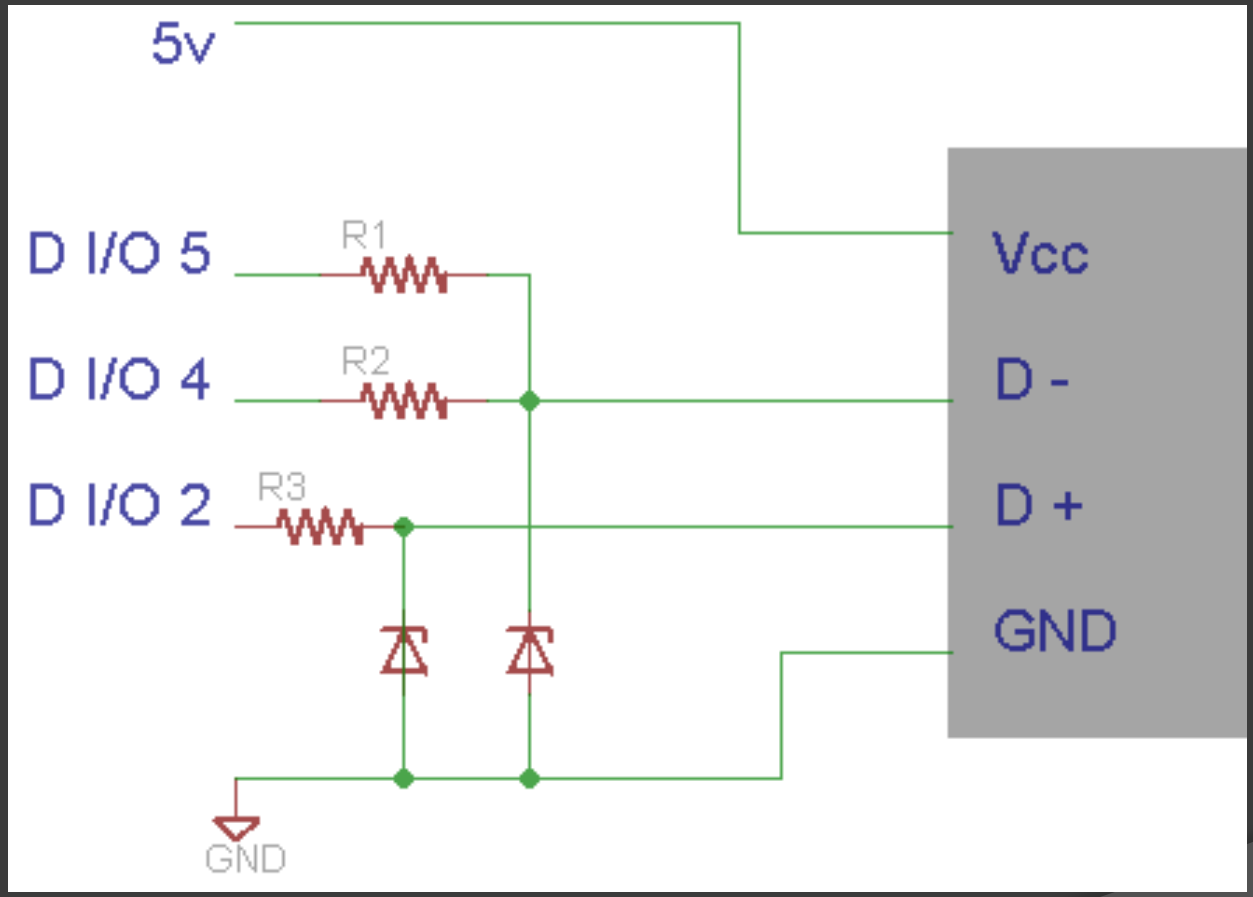
USB HID emulator

- Based on the AVR project V-USB
- simple circuit, 3 resistors and 2 diodes
- any computer thinks a USB keyboard has been connected, no drivers needed.
- can be modified to emulate other USB devices (mouse, joystick)

Security Uses

- this device could be used to try and break out of a kiosk mode by trying every possible combination of keyboards.
- connect it to a computer while the user is away, then after they come back and log in you could distract the user and have it issue commands while the user isn't paying attention (i.e. commands to create a user).

Schematic



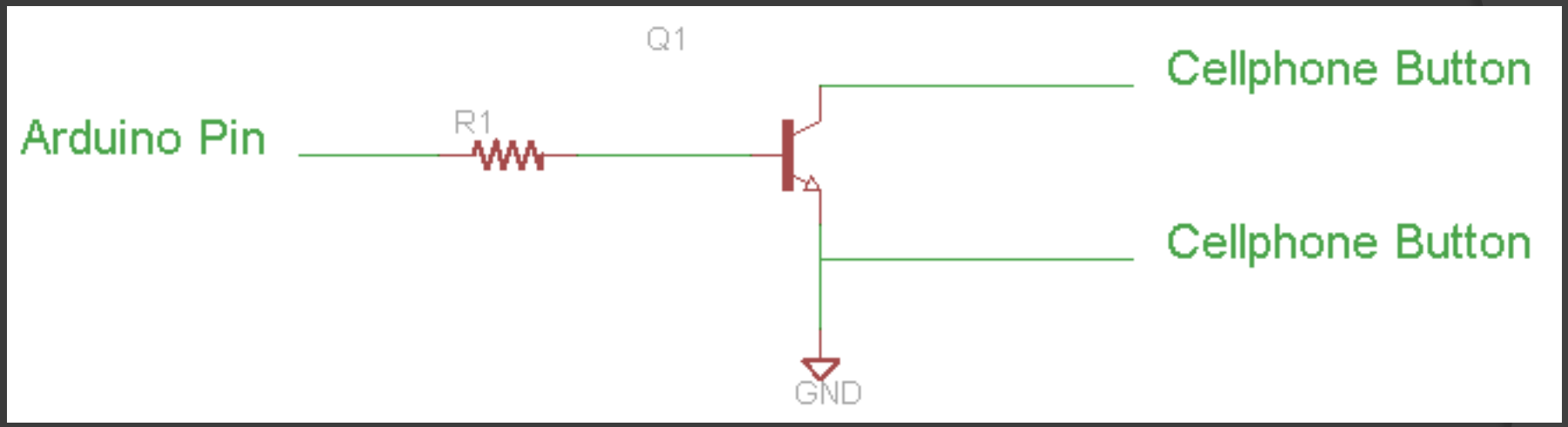
Code

```
//clear built in timers
TIMSK0&=!(1<<TOIE0);
cli();

//connect to computer
usbDeviceConnect();

//send keystrokes
UsbKeyboard.sendKeyStroke(KEY_A);
UsbKeyboard.sendKeyStroke(KEY_B);
```

Integrate with Cellphone

- Solder transistors to the keypad of a cheap prepaid cell phone, connect the transistors to the Arduino (or a shift register connected to the Arduino) and you can control the phone from the Arduino.
- A typical Arduino can't do DTMF decoding, but you can get an IC to do it
- This device could be used for interacting with touch tone phone systems, remote controlling devices, or sending data out of band.
- An alternative to this is to buy a cell phone module and attach it to an Arduino.

Schematic



Code

- Same as blinking an LED, just make the pin high that's connected to the key you want to press

- digitalWrite(ledPin, HIGH);

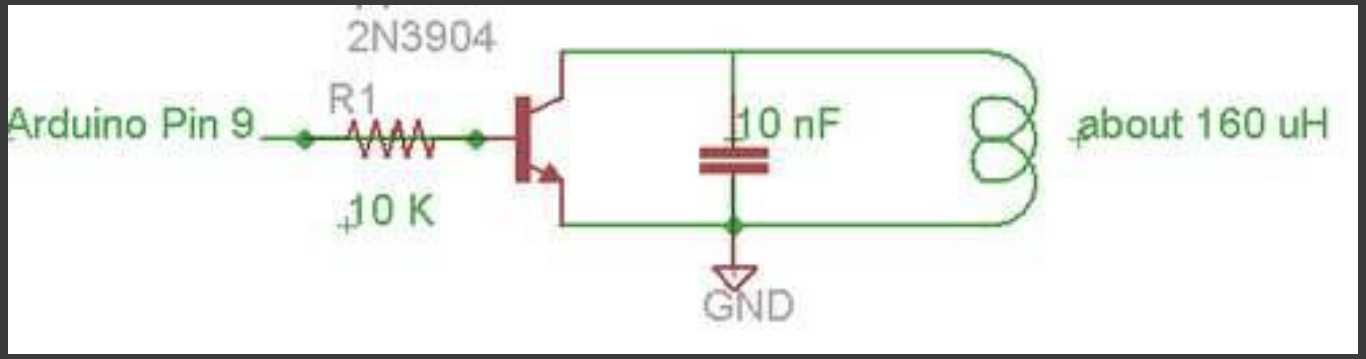
Other Uses for this Technique

- You can use this technique for almost anything you want to automate button pushes on (and you don’t mind soldering to the device).

RFID tag spoofer – stupid simple

- Uses a 10 K Ohm resistor, a transistor, a 10 nF capacitor, and a spool of wire from Radio Shack.
- The code is about 20 lines long and very simple

Schematic



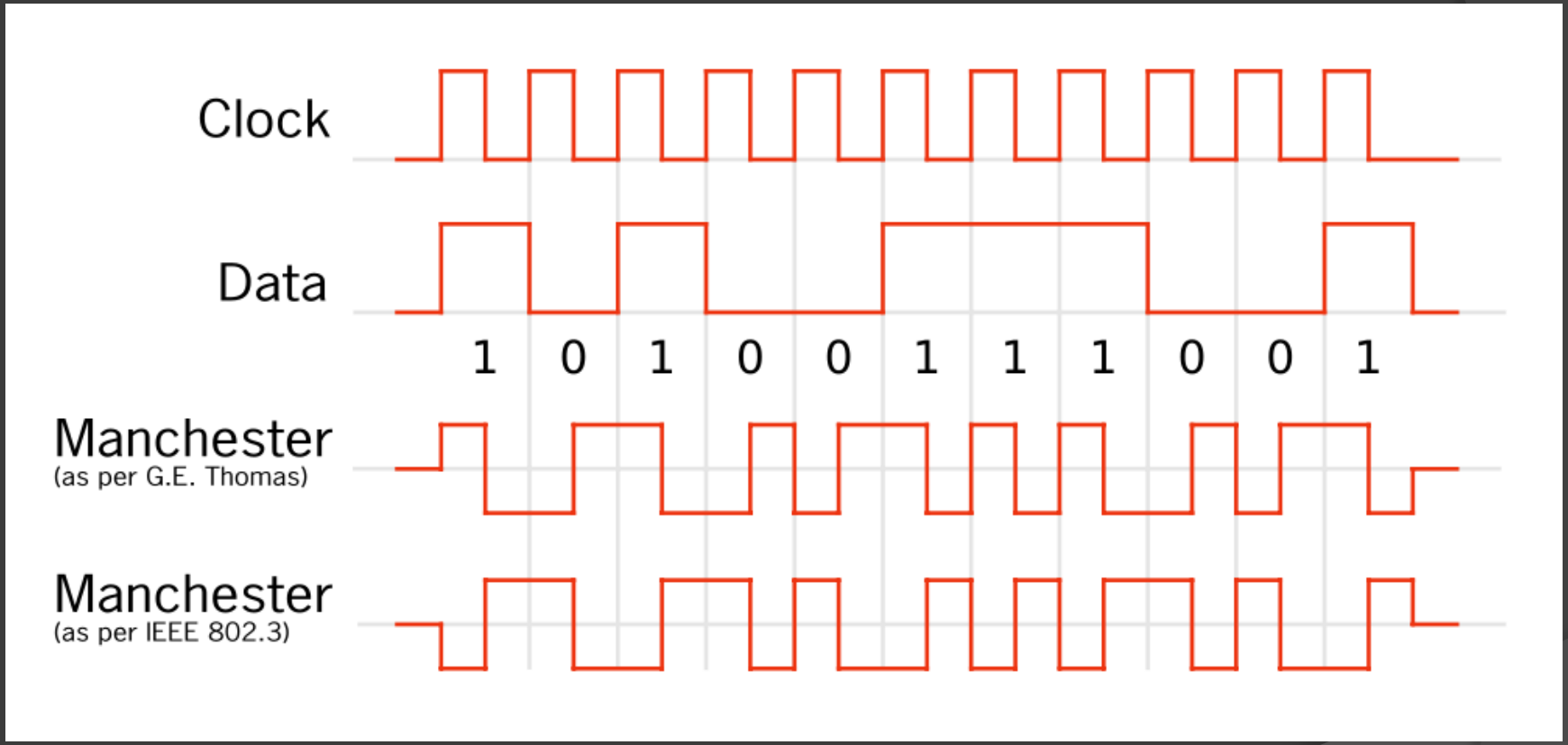
Setup

```
int coil_pin = 9;

void setup()
{
  //Set pin as output
  pinMode(coil_pin, OUTPUT);

  //Start it as low
  digitalWrite(coil_pin, LOW);
}
```

Main Loop Code

```
void loop()
{
  //this is the card data we're spoofing.  It's basically 10 hex F's
  int data_to_spoof[64] = {1,1,1,1,1,1,1,1,1, 1,1,1,1,0 ,1,1,1,1,0,
      1,1,1,1,0 ,1,1,1,1,0, 1,1,1,1,0 ,1,1,1,1,0, 1,1,1,1,0 ,1,1,1,1,0,
      1,1,1,1,0 ,1,1,1,1,0, 0,0,0,0,0};
  for(int i = 0; i < 64; i++)
  {
    set_pin_manchester(0, data_to_spoof[i]);
    delayMicroseconds(256);

    set_pin_manchester(1, data_to_spoof[i]);
    delayMicroseconds(256);
  }
}
```

Manchester Encoding



*Source Wikipedia Manchester Code

Manchester Code

```
void set_pin_manchester(int clock_half, int signal)
{
  //manchester encoding is xoring the clock with the signal
  int man_encoded = clock_half ^ signal;

  //if it's 1, set the pin LOW (this will tune the antenna and the reader sees this as a high
       signal)
  //if it's 0, set the pin to HIGH  (this will detune the antenna and the reader sees this as a
       low signal)
  if(man_encoded == 1)
  {
     digitalWrite(coil_pin, LOW);
  }
  else
  {
    digitalWrite(coil_pin, HIGH);
  }
}
```

Combination Lock brute forcer

- Stepper motor controlled by the Arduino turns the dial
- Servo tries to open the lock (an actuator would work as well)
- A laser pointing to a photoresitor, when the dial hits 0 the beam is broken, this verifies there’s no slipping when turning the dial.
- This device tries every possible combination until it is able to open the lock.
- For locks where there's a known algorithm to open them (like Mater Locks) it open the lock based on the algorithm

Combining devices

- The real power comes in combining several devices.
- Examples
  - RFID reader that broadcasts what it reads using an Xbee

Alternatives to Typical Arduinos

- Maple (ARM based Arduino)
  - more powerful than a typical Arduino (sound and video)
- Butterfly Uno with AVR emulator
  - Run Arduino stuff on an FPGA
- Teensy
  - AVR device with USB capabilities that can use Arduino sketches (even though it appears not all of the functionality of the board is available unless you use C).
- Arduino Megas
  - more memory and inputs and output
- Bare bone or Minimalist Arduinos
  - cheaper but probably doesn't have a USB connector (FTDI is needed to code it)

More Alternatives

- Parallax Propeller
  - Has 8 cogs for parallel computing
- AVR (no Arduino stuff)
  - cheap, code directly in C
- Microchip PIC
  - cheap, code in C or assembly
- Freescale DSC
  - used on DEFCON badges, CodeWarrior pretty easy to use especially with Processor Expert