

# ULTRA LOW COST HACKING

Rodrigo Muñoz Lazo

Agustín Salas Fernández

- 1. EQUIPO DEL PROYECTO**
- 2. PRESUPUESTO PARA JUGAR**
- 3. COMPARATIVA VERSIÓN EMPRESARIAL VS LOW COST**
- 4. REVISIÓN DE 4 EJEMPLOS**
- 5. CONCLUSIONES**
- 6. TRABAJO FUTURO**

# Equipo de investigación en Academias IT.

### Rodrigo Muñoz Lazo

- Ingeniero Electrónico – Universidad Tecnológica Metropolitana.
- Magister en Ingeniería Electrónica – Universidad de Santiago de Chile.
- Asesor Academias IT – Universidad Tecnológica de Chile INACAP.



### Agustín Salas Fernández

- Ingeniero Informática – Universidad Católica de Valparaíso.
- Magister en Ingeniería Informática – Universidad Católica de Valparaíso.
- Asesor Academias IT – Universidad Tecnológica de Chile INACAP.



# VIDEO REFERENCIA

## PRESUPUESTO PARA JUGAR

Que puedo  
comprar con  
\$2.200 CLP?



1 Espresso



1 Kross



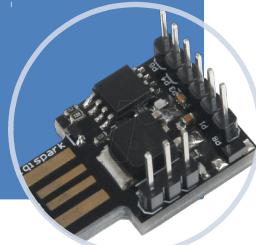
11 Sopaipillas

## PRESUPUESTO PARA JUGAR

Es posible implementar una herramienta de hacking por \$2.200 CLP?

- \$2.200

DigiSpark



- \$3.000

Despacho

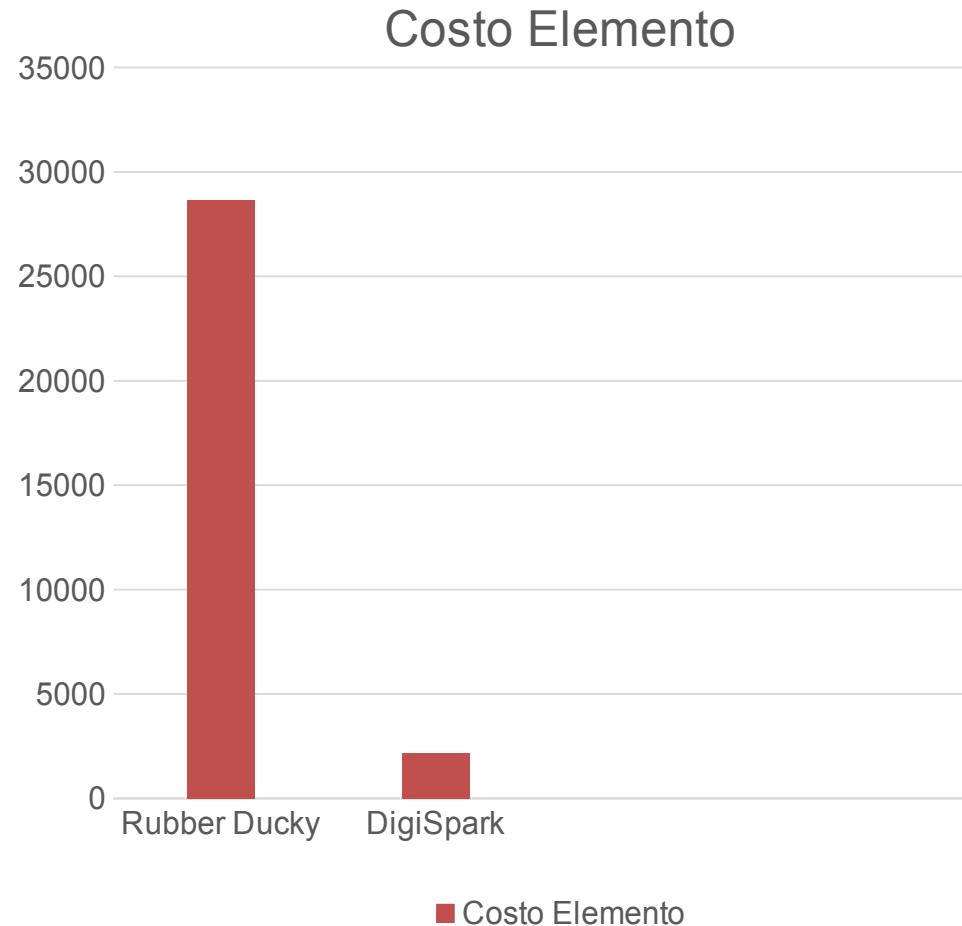


- \$400

Sopaipilla



# PRESUPUESTO PARA JUGAR



Rubber Ducky = \$29.280.-

DigiSpark = \$2.200.-

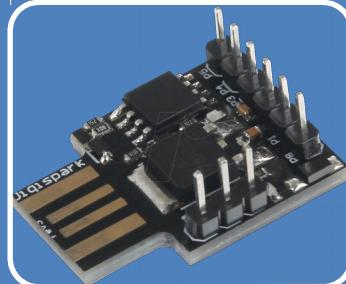
**Con el costo de un Rubber Ducky, se pueden comprar 13 DigiSpark.**

# ELEMENTOS NECESARIOS PARA EL ATAQUE



## IDE de Desarrollo

- Instalación de Drivers
- Configuración de tarjeta
- Programación C++



## DigiSpark / ATTiny85

- Microcontrolador programable
- Opcional: 2 Sopaipillas!



## Imaginación

- Desarrollo del código de ataque.
- Ingeniería social para despliegue.

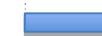
# METODO DE TRABAJO RUBBER DUCKY VS DIGIQUAK



Ducky Script



Duckencode.jar

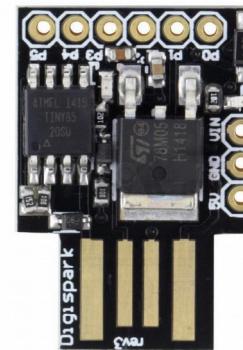


```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // initialize the LCD
  lcd.begin(16, 2);
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.print("Hello, world!");
  delay(1000);
}
```



# COMPARATIVA VERSION EMPRESARIAL VS LOW COST



## Rubber Ducky

60 MHz  
32-bit CPU  
AT32UC3B1256

Micro SD

Botón Replay

GPIO y DFU

## DigiQuack

20 MHz  
ATTiny85 8bit

512-Byte SRAM  
512-Byte ROM

**digiQuack**

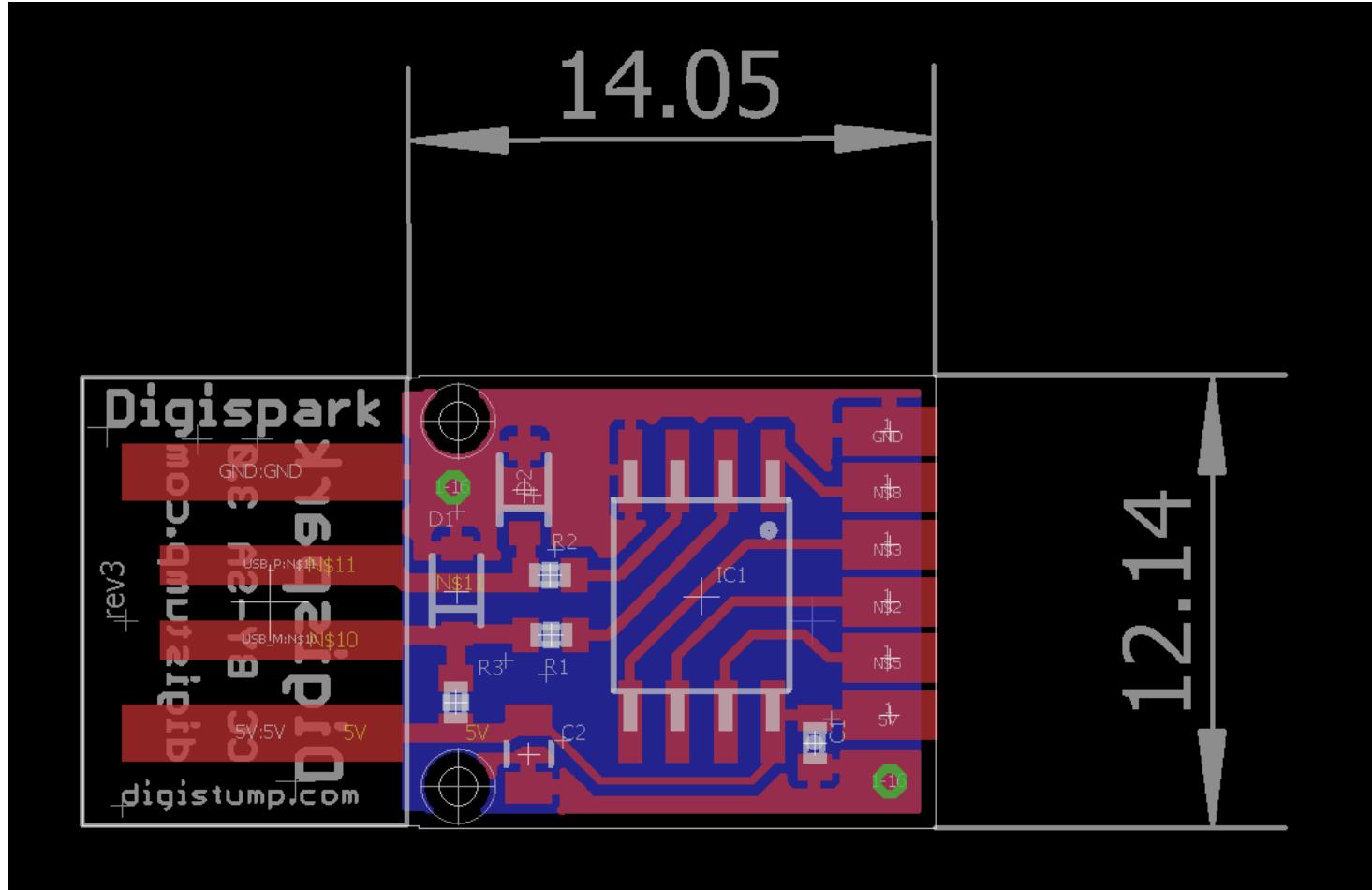
## COMPARATIVA DE MICROCONTROLADORES

uC	CPU Bits	Mem. Prog.	Mem. RAM	Mem. EEPROM	FREC. MAX.	USB	PRECIO*
AT32UC3B1256	32	256KB	32KB	NO	60MHz	SI	6.54 USD
ATTiny85	8	8KB	512B	512 Bytes	20MHz	NO	1.23 USD

\* Digikey ([www.digikey.com](http://www.digikey.com))

Slot Sdcard ~1.31USD  
 Conector USB ~1.24USD  
 Boton ~1USD

## EJEMPLO DE DISMINUCIÓN DE TAMAÑO



# EJEMPLO 1/4:

# HOLA MUNDO

# EJEMPLO 2/4:

## LISTADO DE DIRECTORIES CON POWERSHELL

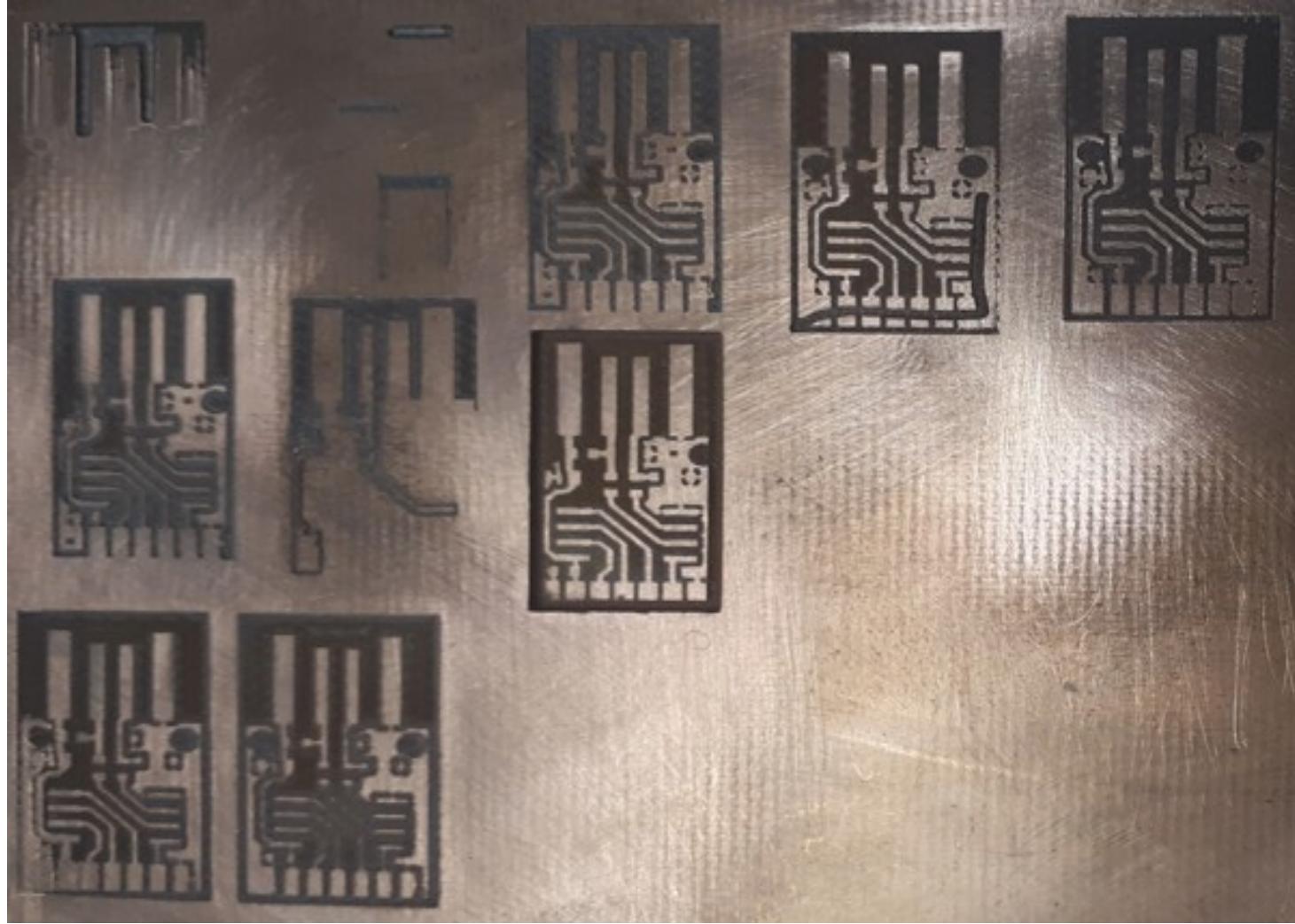
# EJEMPLO 3/4:

## DESCARGA DESDE WEB <3 POWERSHELL

# EJEMPLO 4/4: TWITTEO AUTOMÁGICO REALLY <3 PYTHON



# TRABAJANDO EN DISMINUIR EL TAMAÑO

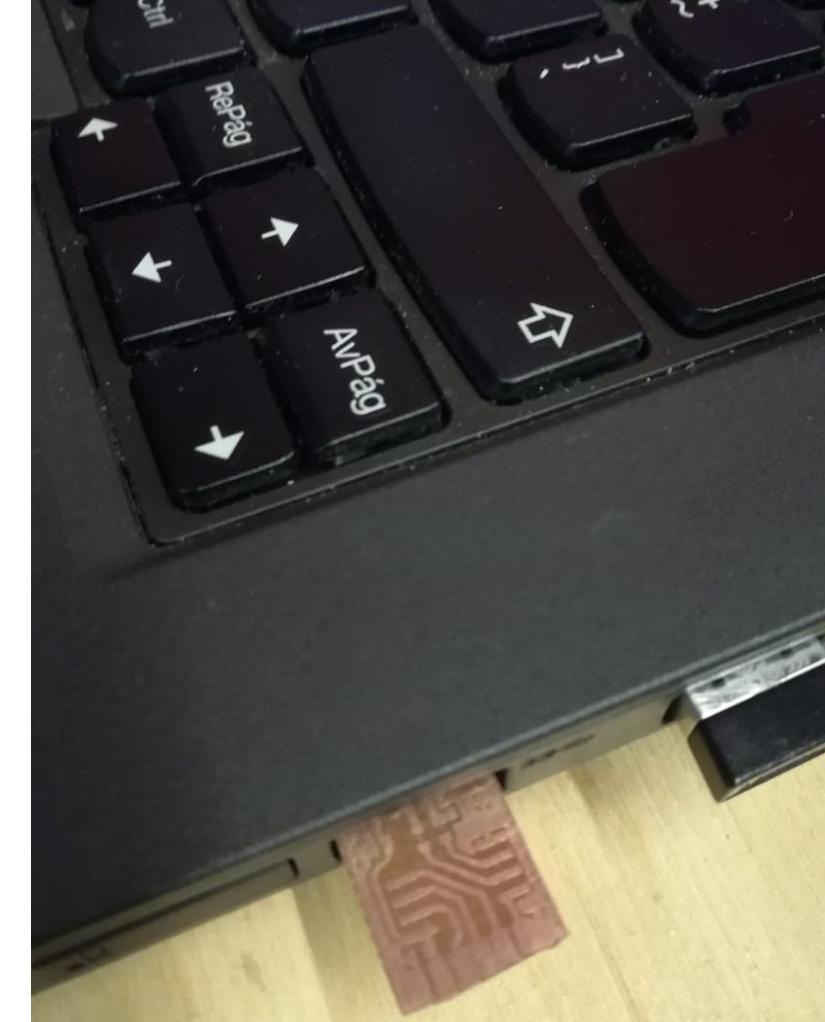


**Múltiples intentos de grabar la placa con un torno CNC.**

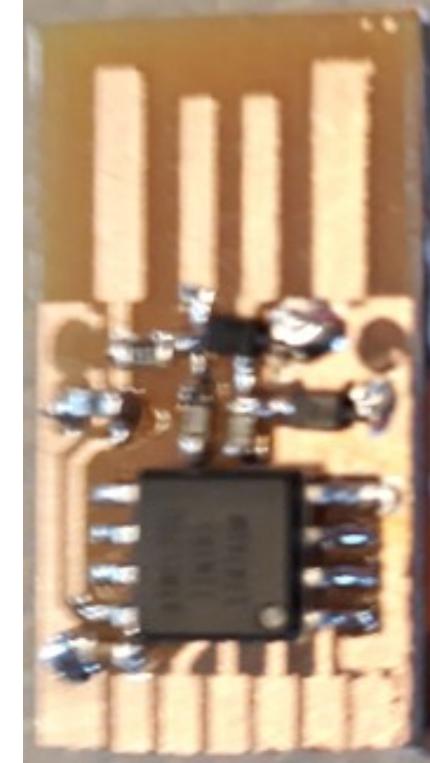
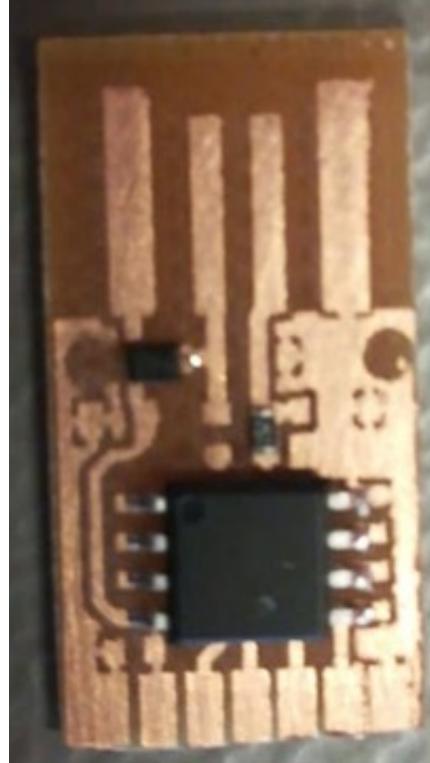
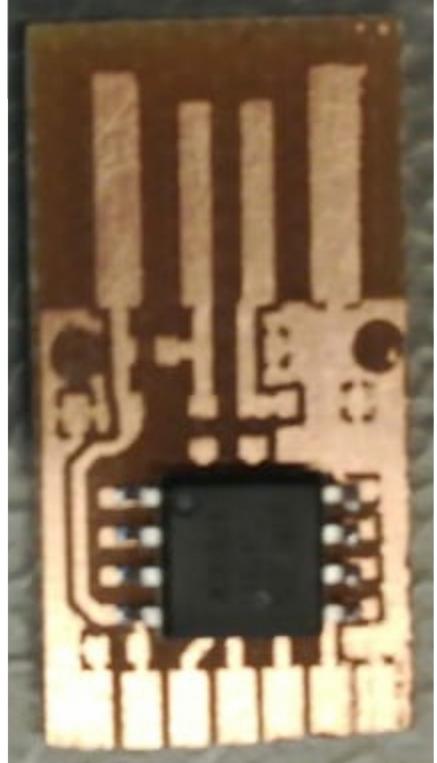
**Proceso iterativo.  
Ensayo / Error.**

## TRABAJANDO EN DISMINUIR EL TAMAÑO

Luego de contar con la placa correctamente grabada, se debe trabajar en ajustar el tamaño y grosor, de manera tal que se sostenga correctamente en el puerto USB.



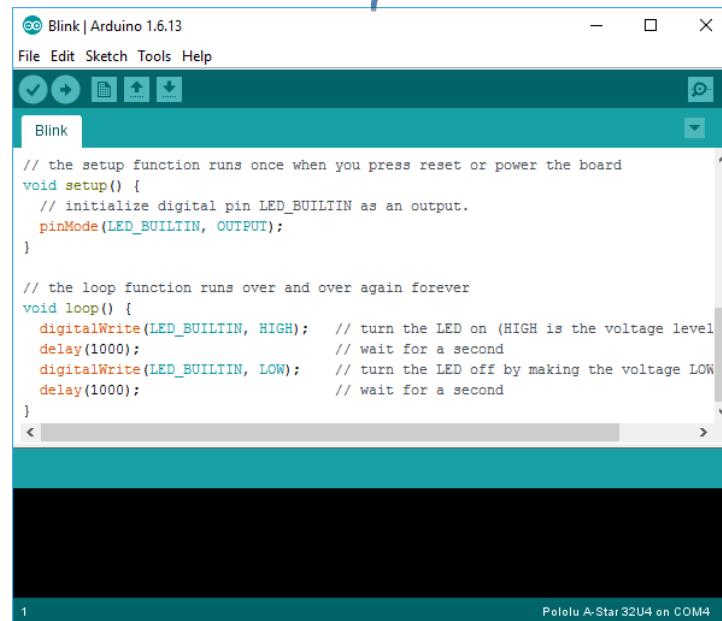
# TRABAJANDO EN DISMINUIR EL TAMAÑO



**Proceso de  
soldado de los  
componentes**

# SI EL ATTINY85 ESTUVIERA NUEVO...

Programar Bootloader



Blink | Arduino 1.6.13

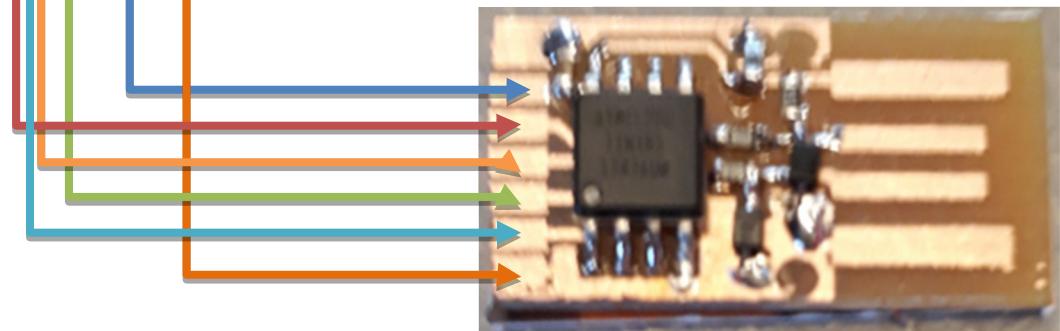
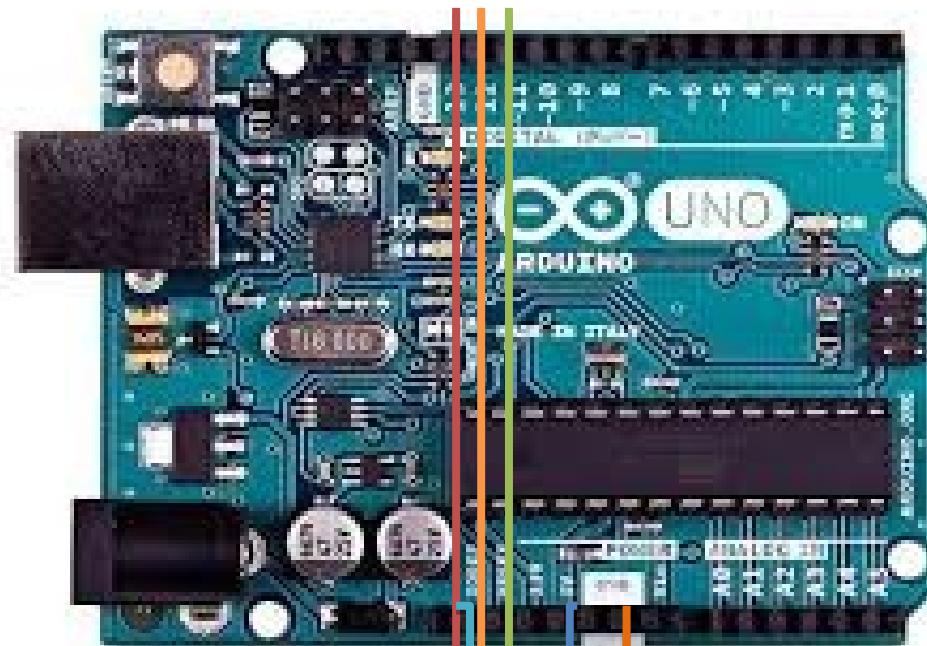
File Edit Sketch Tools Help

Blink

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

Polello A-Star 32U4 on COM4



Arduino  
Como  
ISP



# CONCLUSIONES

- El < costo => Experimentar.
- DigiSpark < Rubber Ducky.
- DigiSpark es de programación realmente fácil.
- DigiSpark se puede camuflar con un case impreso en 3D
- Se puede hackear el BootLoader
- DigiSpark es tremadamente barato

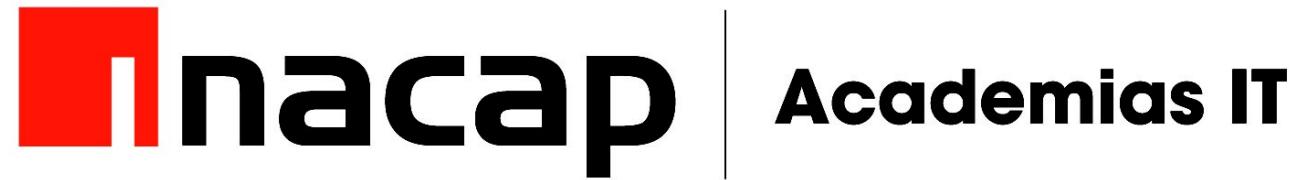


- Probar performance de RD y DS en tareas específicas.
- Realizar exfiltración mediante RF.
- Realizar conexión reversa.
- **Salvar los problemas típicos como:**
  - Tiempo de respuesta o Delay
  - Salvar problemas como distribución de teclados
  - Bajada de AV para ejecución de Mimikatz

# DEMO SORPRESA

}:)

# AGRADECIMIENTOS



# PREGUNTAS?



@agustin\_salas\_f