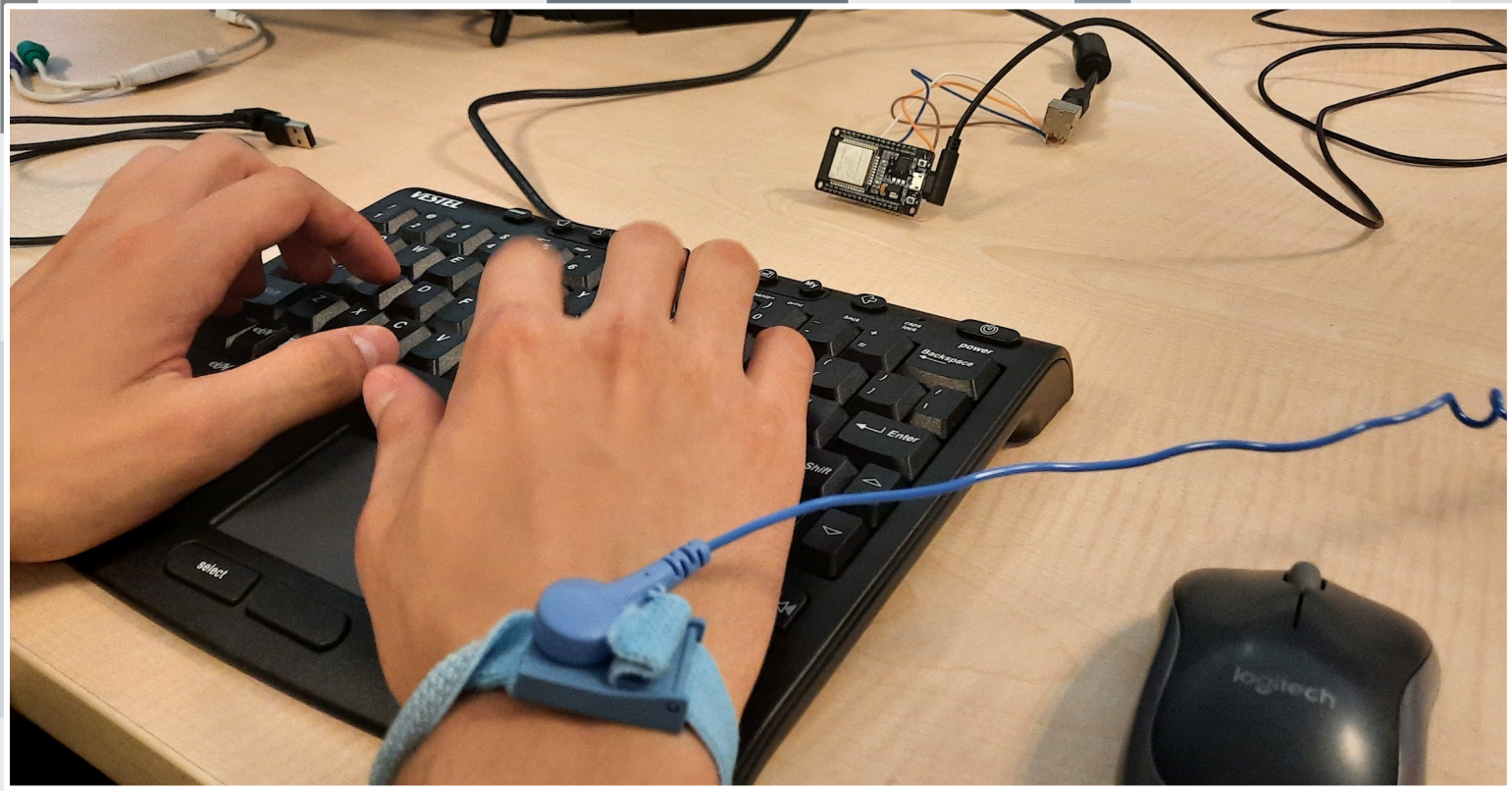


Bluetooth Klavye Profili ile Kablosuz Hata Ayıklama

Denis Davidoglu,
İTÜ Bilgisayar Mühendisliği
Ar-Ge Gömülü Yazılım Stajyeri
Temmuz 2023

Giriş ve Amaç

- Satılmış arızalı cihazlara mühendislerin doğrudan müdahale etmesi zordur.
- Seri üretimindeki cihazların JTAG, SWD, UART gibi hata ayıklama portları yoktur.
- Kablosuz Bluetooth portu mümkündür.
- Cihaz, müşterinin telefonunda klavye yetkisini alarak e-posta veya WhatsApp’tan hata mesajlarını otomatik olarak yazabilir.

Bluetooth

Yığın yapısı, İnsan Arayüz Cihazı profili, yüksek seviyeli API

PS/2

Protokol özellikleri, sürücüsü

ESP32

ESP-IDF, kütüphaneleri, örnek kodları

USB

Rapor haritası, tarama kodları

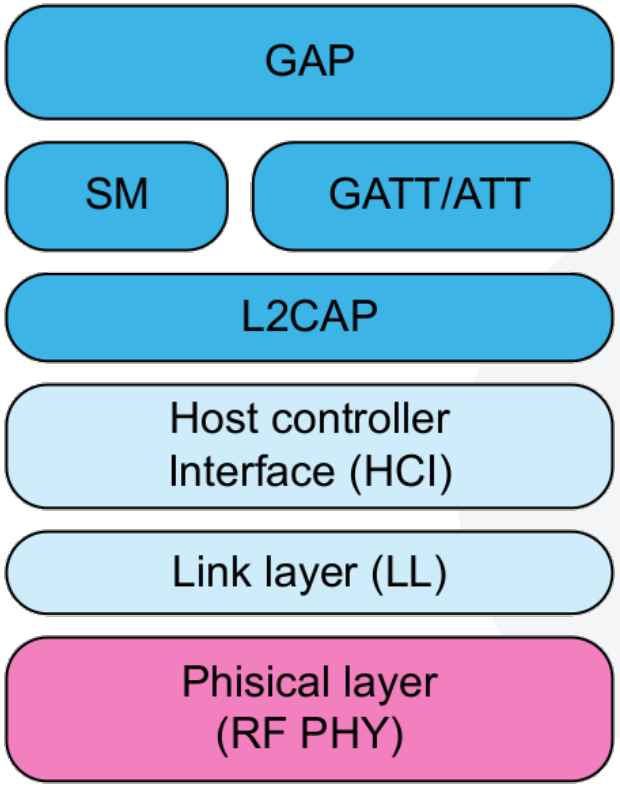
Elektronik

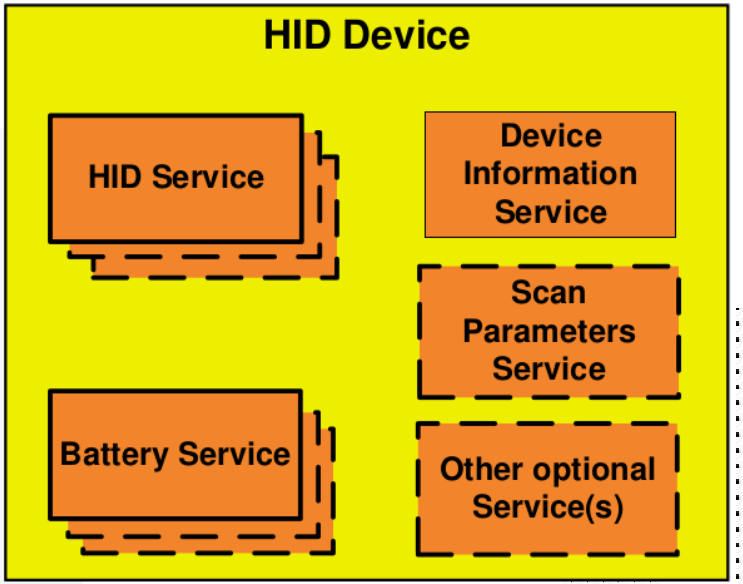
Donanım, prototipler

FreeRTOS

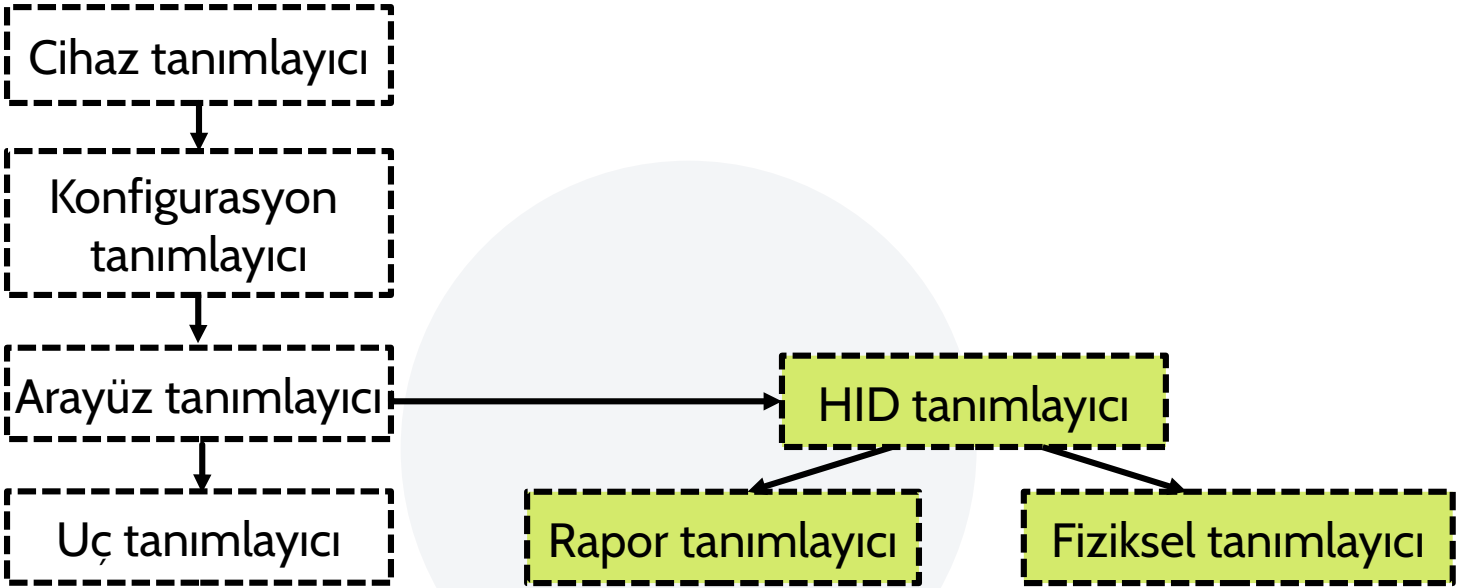
Kesmeler, task uyandırma, sıra veri tipi

Bluetooth Yığını





USB HID sınıfı



HID rapor haritası

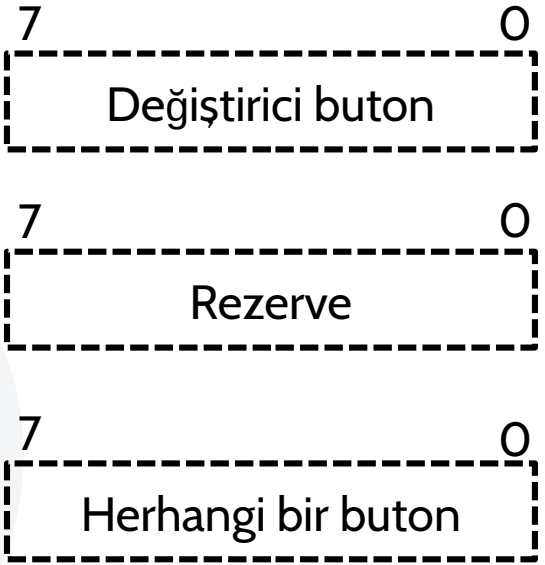
```
const unsigned char keyboardReportMap[] = {
  0x05, 0x01, /* Usage Page (Generic Desktop), */
  0x09, 0x06, /* Usage (Keyboard), */
  0xA1, 0x01, /* Collection (Application), */
    0x05, 0x07, /* Usage Page (Key Codes); */
    0x19, 0xE0, /* Usage Minimum (224), */
    0x29, 0xE7, /* Usage Maximum (231), */
    0x15, 0x00, /* Logical Minimum (0), */
    0x25, 0x01, /* Logical Maximum (1), */
    0x75, 0x01, /* Report Size (1), */
    0x95, 0x08, /* Report Count (8), */
    0x81, 0x02, /* Input (Data, Variable, Absolute), */
    0x95, 0x01, /* Report Count (1), */
    0x75, 0x08, /* Report Size (8), */
    0x81, 0x01, /* Input (Constant), */
    0x95, 0x05, /* Report Count (5), */
    0x75, 0x01, /* Report Size (1), */
    0x05, 0x08, /* Usage Page (Page# for LEDs), */
    0x19, 0x01, /* Usage Minimum (1), */
    0x29, 0x05, /* Usage Maximum (5), */
    0x91, 0x02, /* Output (Data, Variable, Absolute), */
    0x95, 0x01, /* Report Count (1), */
    0x75, 0x03, /* Report Size (3), */
    0x91, 0x01, /* Output (Constant), */
    0x95, 0x06, /* Report Count (6),*/
    0x75, 0x08, /* Report Size (8), */
    0x15, 0x00, /* Logical Minimum (0), */
    0x25, 0x65, /* Logical Maximum(101), */
    0x05, 0x07, /* Usage Page (Key Codes), */
    0x19, 0x00, /* Usage Minimum (0), */
    0x29, 0x65, /* Usage Maximum (101), */
    0x81, 0x00, /* Input (Data, Array), */
  0xC0 /* End Collection */
};
```

HID rapor haritası

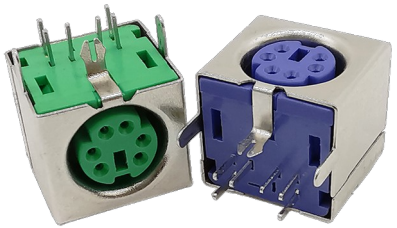
```
0x75, 0x01, /* Report Size (1), */
0x95, 0x08, /* Report Count (8), */
0x81, 0x02, /* Input (Data, Variable,
```

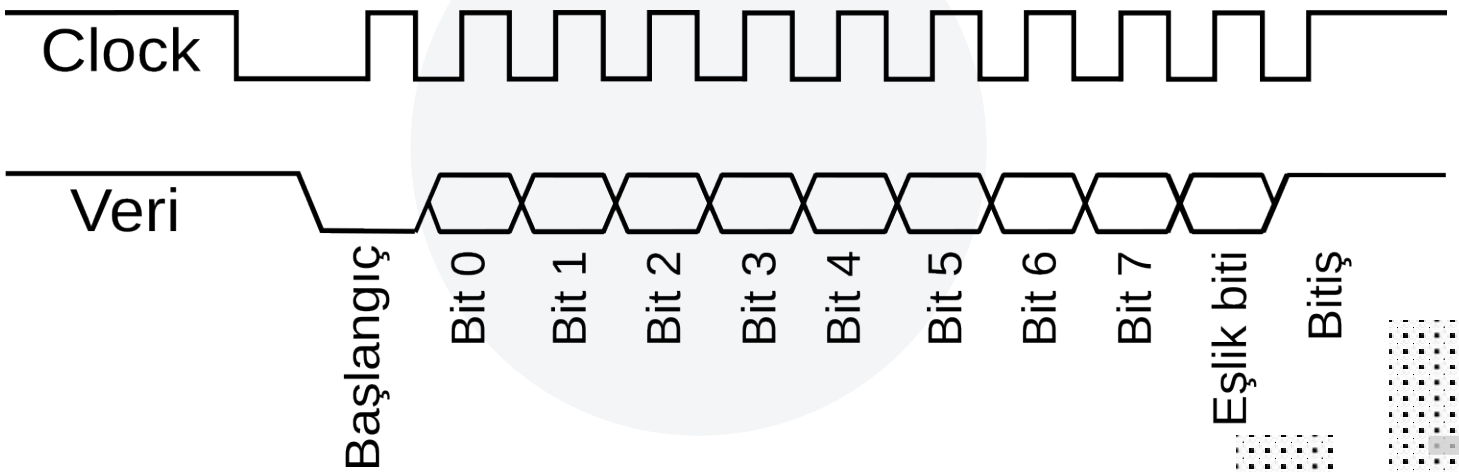
```
0x95, 0x01, /* Report Count (1), */
0x75, 0x08, /* Report Size (8), */
0x81, 0x01, /* Input (Constant), */
```

```
0x95, 0x01, /* Report Count (1) */
0x75, 0x08, /* Report Size (8), */
 ...
0x81, 0x00, /* Input (Data, Array), */
```

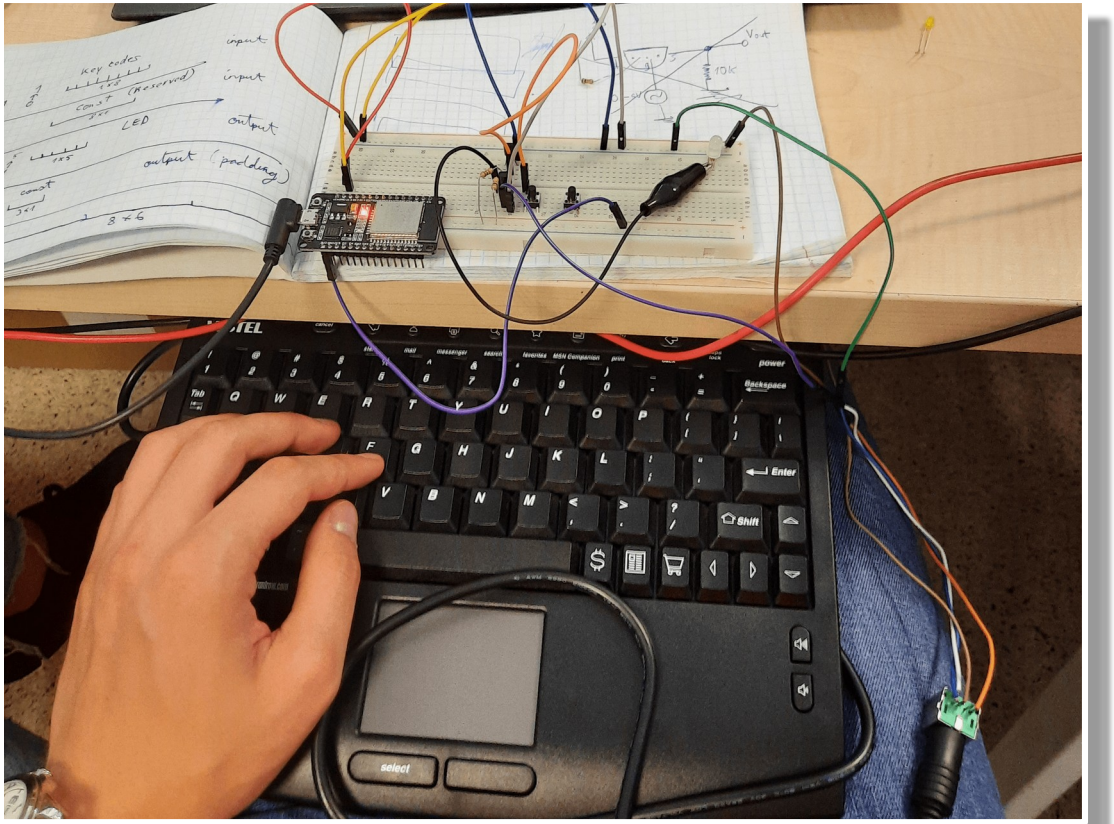


PS/2 protokolü





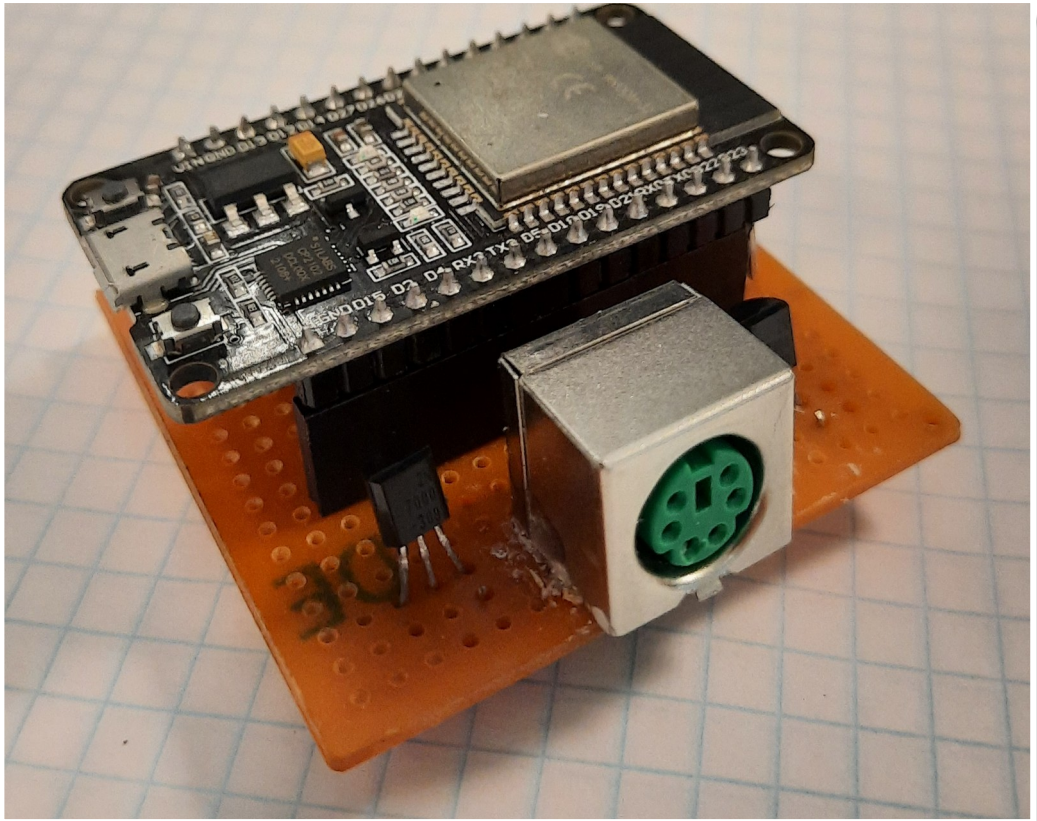
Elektronik prototipleri



00 breadboard

5V ile çalışan klavyenin sinyalleri, mosfetlerden yapılmış voltaj düşürücüden geçerek LEDi yaktı

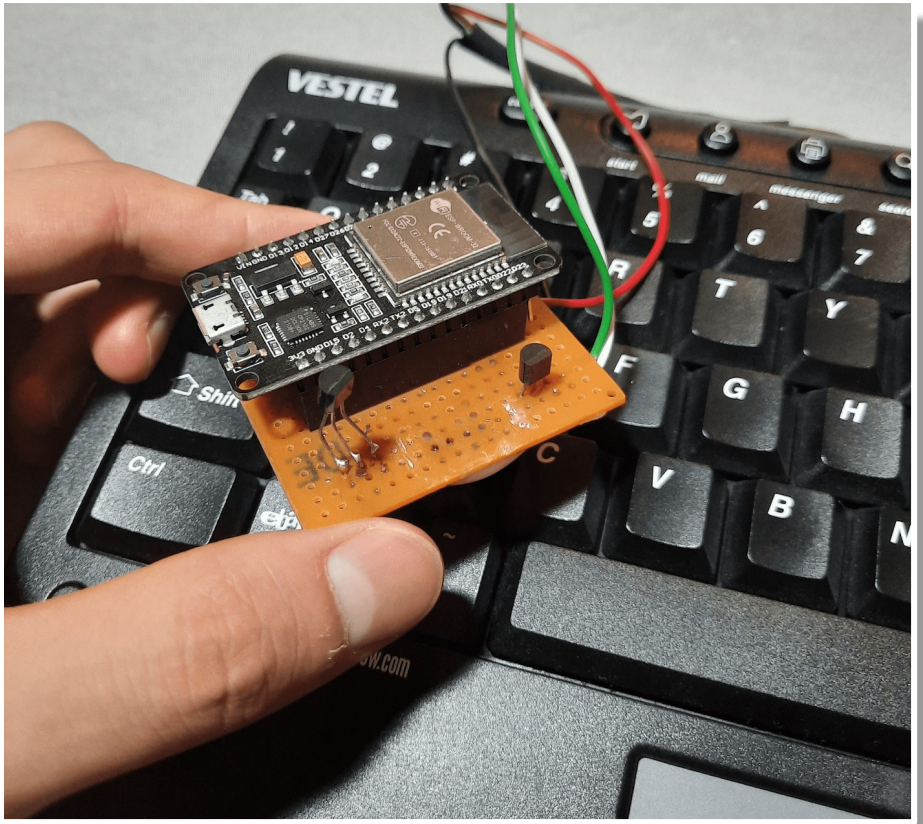
Elektronik prototipleri



01 PS/2 portu

PS/2 portu, voltaj düşürücü ve ESP32, delikli plakete lehimlendi

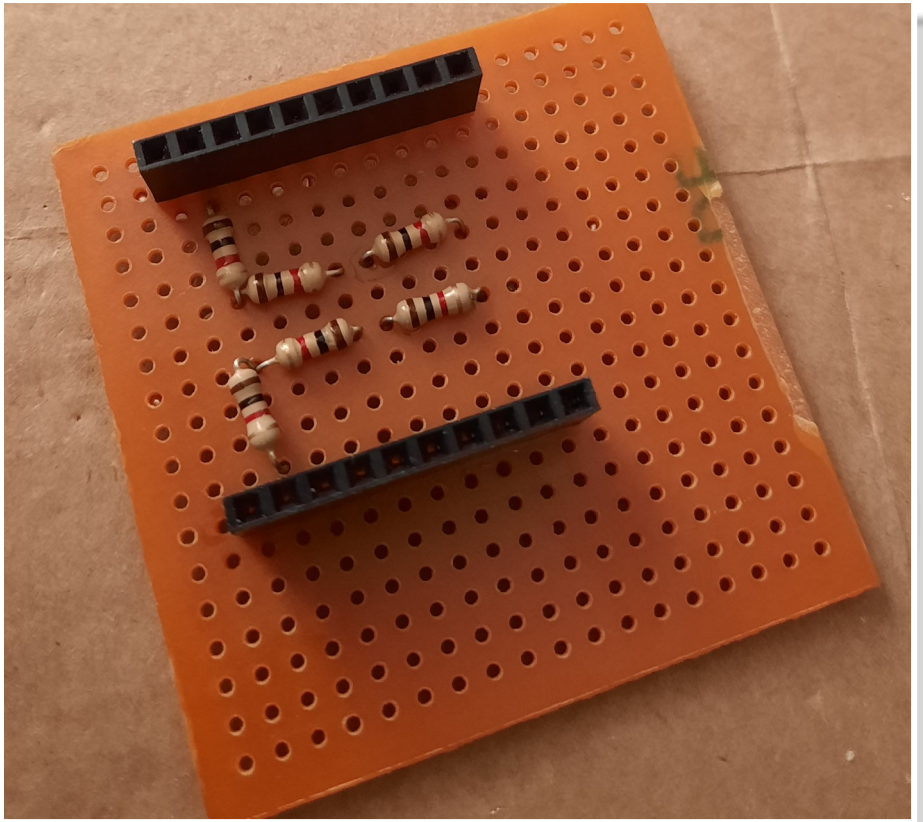
Elektronik prototipleri



02 kaba lehim

Temassızlık yapan port çıkarıldı, klavye direkt lehimlendi, bir mosfet değiştirildi

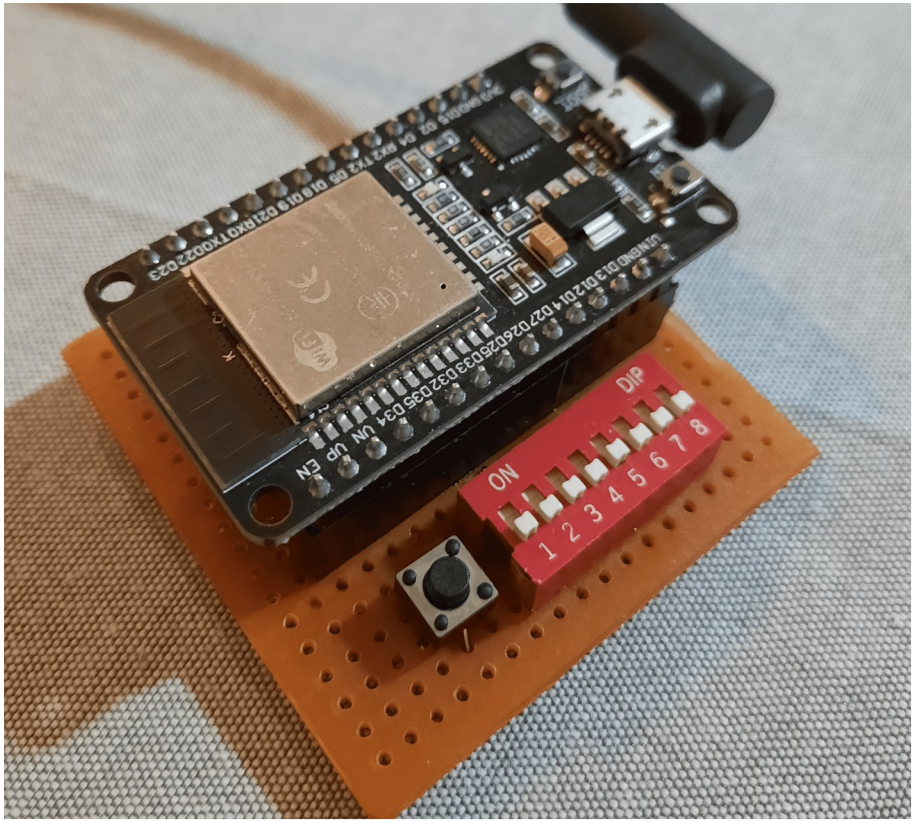
Elektronik prototipleri



03 voltaj bölücü

Mosfet devresi, 1kΩ ve 2kΩ resistörlü voltaj bölücü ile değiştirildi

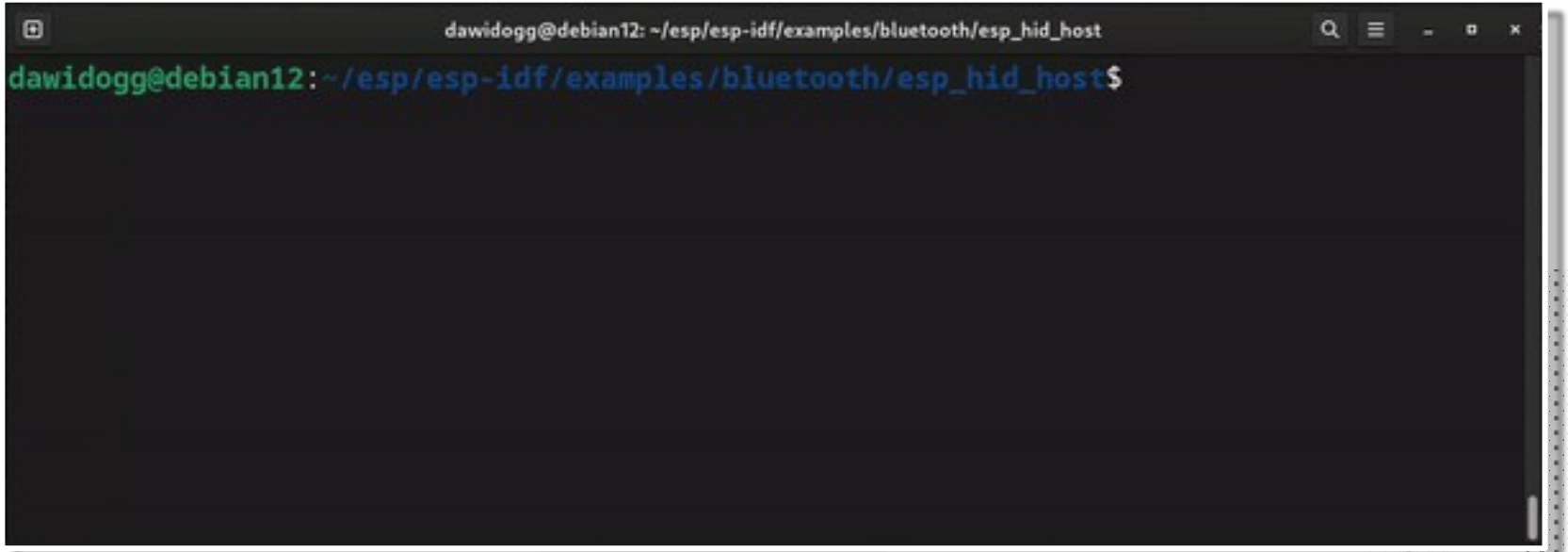
Elektronik prototipleri



04 ASCII seçici

Klavyeden vazgeçilip, 8’li switch ve buton ile ASCII seçici yapıldı

ESP32 ile C kodlama



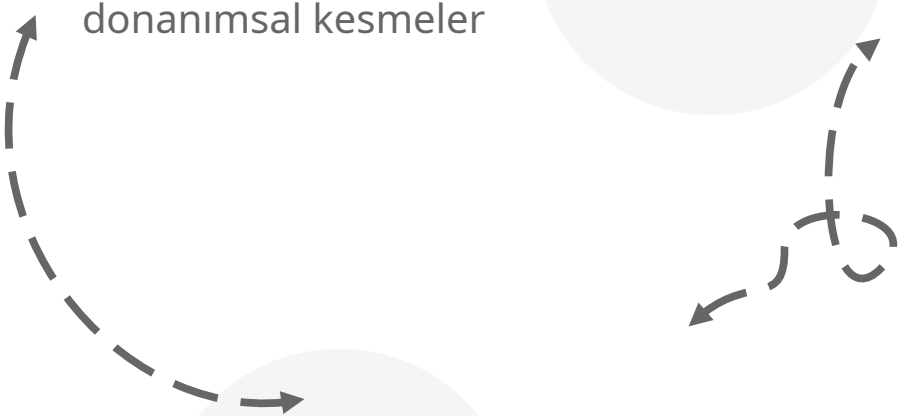
ESP-IDF kütüphaneleri

driver/gpio.h

Pinlerden dijital sinyal okuma, donanımsal kesmeler

esp_bt.h

Bluetooth yüksek seviyeli API



freertos/FreeRTOS.h

Paralel tasklar, erteleme, uyandırma, sıra veri yapısı

esp_hidd.h

İnsan Arayüzü Cihazı fonksiyonları

Tablolar

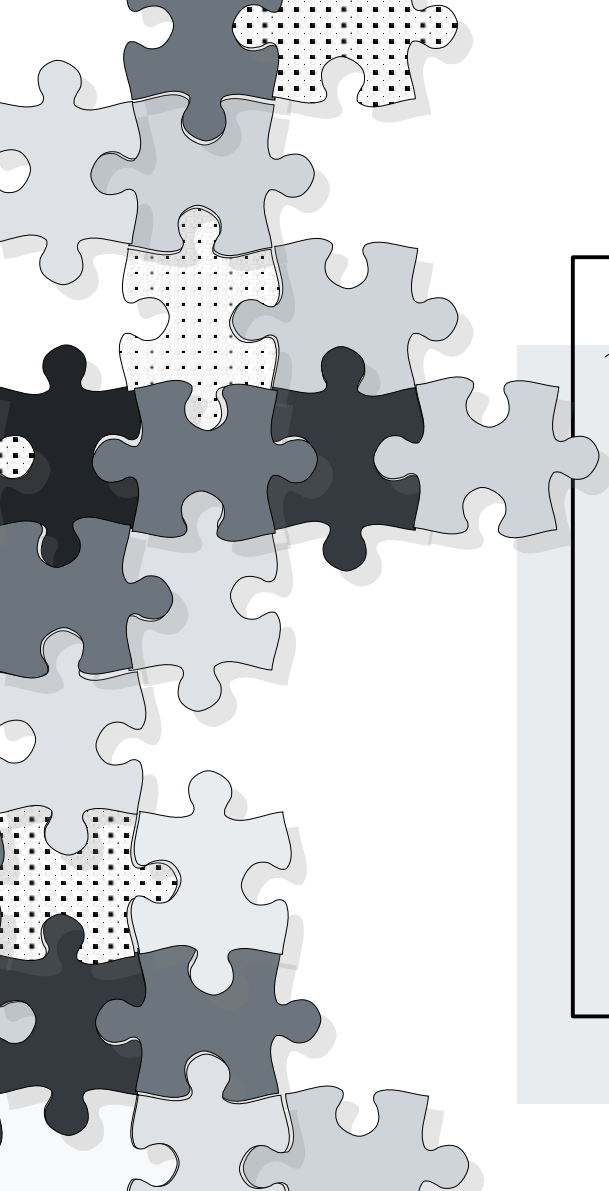
```
static const uint8_t usb_scancodes[] = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x2a, 0x00, 0x28, 0x00, 0x00, 0x00, 0x00, 0x00
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
//       !     "     #     $     %     &     '     (     )     *     +     ,     -     .     /
0x2c, 0x1e, 0x35, 0x20, 0x21, 0x22, 0x23, 0x1f, 0x25, 0x26, 0x2d, 0x21, 0x31, 0x2e, 0x38, 0x24
// 0     1     2     3     4     5     6     7     8     9     :     ;     <     =     >     ?
0x27, 0x1e, 0x1f, 0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x38, 0x31, 0x35, 0x27, 0x1e, 0x2d
// @     A     B     C     D     E     F     G     H     I     J     K     L     M     N     O
0x14, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x34, 0x0d, 0x0e, 0x0f, 0x10, 0x11, 0x12
// P     Q     R     S     T     U     V     W     X     Y     Z     [     \     ]     ^     _
0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x25, 0x2d, 0x26, 0x20, 0x2e
// `     a     b     c     d     e     f     g     h     i     j     k     l     m     n     o
0x34, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0a, 0x0b, 0x34, 0x0d, 0x0e, 0x0f, 0x10, 0x11, 0x12
// p     q     r     s     t     u     v     w     x     y     z     {     |     }     ~
0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x24, 0x2e, 0x27, 0x30, 0x00
};
```

```
static const uint16_t usb_modifier_shift[] = {
// 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
  0x0020,
// 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0x0000,
//   ! " # $ % & ' ( ) * + , - . /
// 0 1 0 0 0 1 1 1 1 1 0 1 0 0 0 1
  0x47d1,
// 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
// 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1
  0x0035,
// @ A B C D E F G H I J K L M N O
// 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0x8000,
// P Q R S T U V W X Y Z [ \ ] ^ _
// 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
  0x0003,
// ` a b c d e f g h i j k l m n o
// 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  0x7fff,
// p q r s t u v w x y z { | } ~
// 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
  0xffe0
};
```

```
static const char usb_scancodes_special[][6] = {
// char, char, null,  key, shift, alt
  {0xc4, 0x9f, '\0', 0x2f,     1,    0}, // ğ
  {0xc3, 0xbc, '\0', 0x30,     1,    0}, // ü
  {0xc5, 0x9f, '\0', 0x33,     1,    0}, // ş
  {0xc4, 0xb1, '\0', 0x0c,     1,    0}, // ı
  {0xc3, 0xb6, '\0', 0x36,     1,    0}, // ö
  {0xc3, 0xa7, '\0', 0x37,     1,    0}, // ç
  {0xc4, 0x9e, '\0', 0x2f,     0,    0}, // Ğ
  {0xc3, 0x9c, '\0', 0x30,     0,    0}, // Ü
  {0xc5, 0x9e, '\0', 0x33,     0,    0}, // Ş
  {0xc4, 0xb0, '\0', 0x34,     0,    0}, // İ
  {0xc3, 0x96, '\0', 0x36,     0,    0}, // Ö
  {0xc3, 0x87, '\0', 0x37,     0,    0}, // Ç
};
```

Fonksiyonlar

- `void hid_send_ascii_char(char);`
- `int hid_send_nonascii_char(char*);`
- `void hid_send_capslock();`
- `void hid_send_string(char*);`
- `void ps2_interrupt_respond(void*);`
- `void ps2_print(void*);`



Kaynakça

[1] STMicroelectronics (2023), Guidelines for Bluetooth® Low Energy stack programming on STM32WB/STM32WBA MCUs, Rev 7.
[2] Bluetooth SIG, Inc. (2011), HID over GATT Profile Specification.
[3] Bluetooth SIG, Inc. (2023), Assigned Numbers.
[4] USB Implementers' Forum (2001), Device Class Definition for Human Interface Devices (HID), Version 1.11.
[5] USB Implementers' Forum (2022), HID Usage Tables for Universal Serial Bus (USB), Version 1.3.
[6] Microsoft Corporation (2004), USB HID to PS/2 Scan Code Translation Table.
[7] Adam Chapweske (2003), The PS/2 Mouse/Keyboard Protocol.
[8] Espressif Systems (Shanghai) Co., Ltd (2023), ESP-IDF API Reference.
[9] Amazon.com, Inc. (2017), Reference Manual for FreeRTOS, version 10.0.0, issue 1.

Dinlediğiniz için teşekkürler!