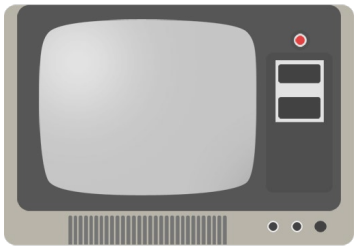
Tandy Assembly
Springfield, OH
9/27/19

Tinkering with the TRS-80's I/O Bus

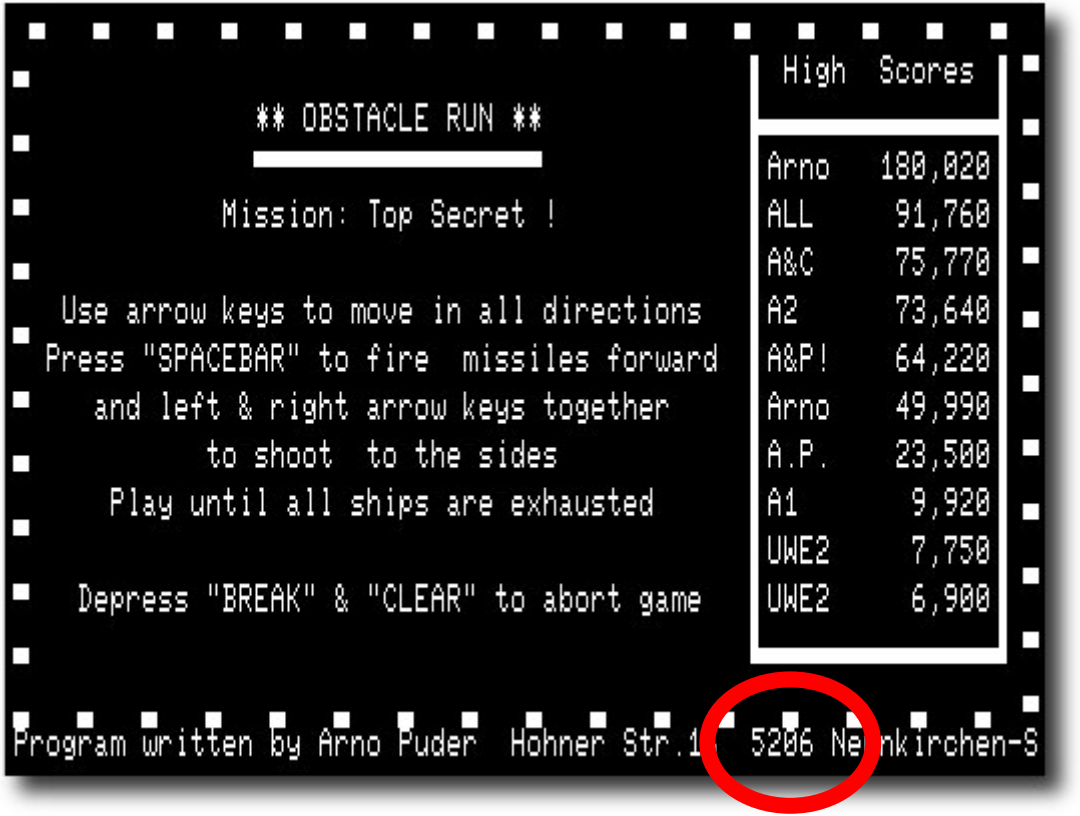
Arno Puder



Outline:

1. My Story
2. TRS-80 I/O Bus
3. Boolean Logic
4. Connecting Arduino with the I/O Bus


** OBSTACLE RUN **
Mission: Top Secret !
Use arrow keys to move in all directions
Press "SPACEBAR" to fire missiles forward
and left & right arrow keys together
to shoot to the sides
Play until all ships are exhausted
Depress "BREAK" & "CLEAR" to abort game
High Scores
Arno 180,020
ALL 91,760
A&C 75,770
A2 73,640
A&P! 64,220
Arno 49,990
A.P. 23,500
A1 9,920
UWE2 7,750
UWE2 6,900
Program written by Arno Puder Hohner Str.16 5206 Neunkirchen-S


** OBSTACLE RUN **
Mission: Top Secret !
Use arrow keys to move in all directions
Press "SPACEBAR" to fire missiles forward
and left & right arrow keys together
to shoot to the sides
Play until all ships are exhausted
Depress "BREAK" & "CLEAR" to abort game
High Scores
Arno 180,020
ALL 91,760
A&C 75,770
A2 73,640
A&P! 64,220
Arno 49,990
A.P. 23,500
A1 9,920
UWE2 7,750
UWE2 6,900
Program written by Arno Puder Hohner Str.1 5206 Neunkirchen-S

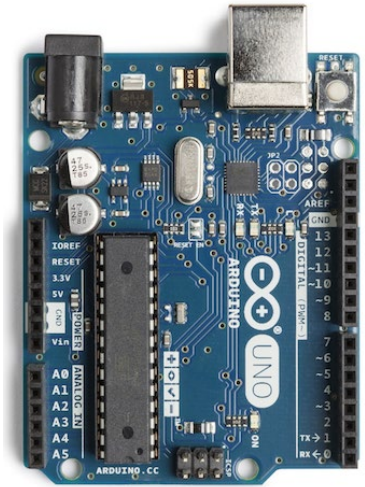
Good Old Days

- TRS-80 Model III
- Manufactured by Radio Shack Corporation
- Released in 1980
- 8-bit Z80 CPU @ 1.7 MHz, 48kB RAM
- Great for tinkering!

Modern Devices are Tinker Resistant!

Arduino

- Open source specification for a programmable microcontroller.
- Started in 2003 in Italy.
- Main purpose is to interface with sensors and actuators via GPIO/I2C and SPI.
- CPUs: Atmel AVR, ARM Cortex, Intel Quark (x86)
- Does not run an operating system!
- Arduino Uno R3:
  - CPU: ATmega328P @ 16 MHz
  - Flash memory: 32 kB
  - SRAM: 2 kB
  - Interface: GPIO, UART, SPI, I2C.
  - Size: 68 x 53 mm
  - Power consumption: 225 mW
  - Arduino Pro Mini: 15 µW - 12 mW
    (4 years on a 9V battery)
- Homepage: https://www.arduino.cc/



TRS-80 I/O Bus

- 50 pin connector; 25 pins are ground.
- Uses TTL (Transistor-Transistor Logic). High = 5V
- 8 bits data bus D0-D7.
- 8 bits for address lines A0-A7.
- Control lines for read, write, wait, interrupts
- BASIC:
  - Input: **`INP(p)`**
  - Output: **`OUT p, n`**
- Z80:
  - **`out (p),n`**
  - **`in r,(p)`**
- I/O bus needs to be enabled first via OUT 236,16

Radio Shack®

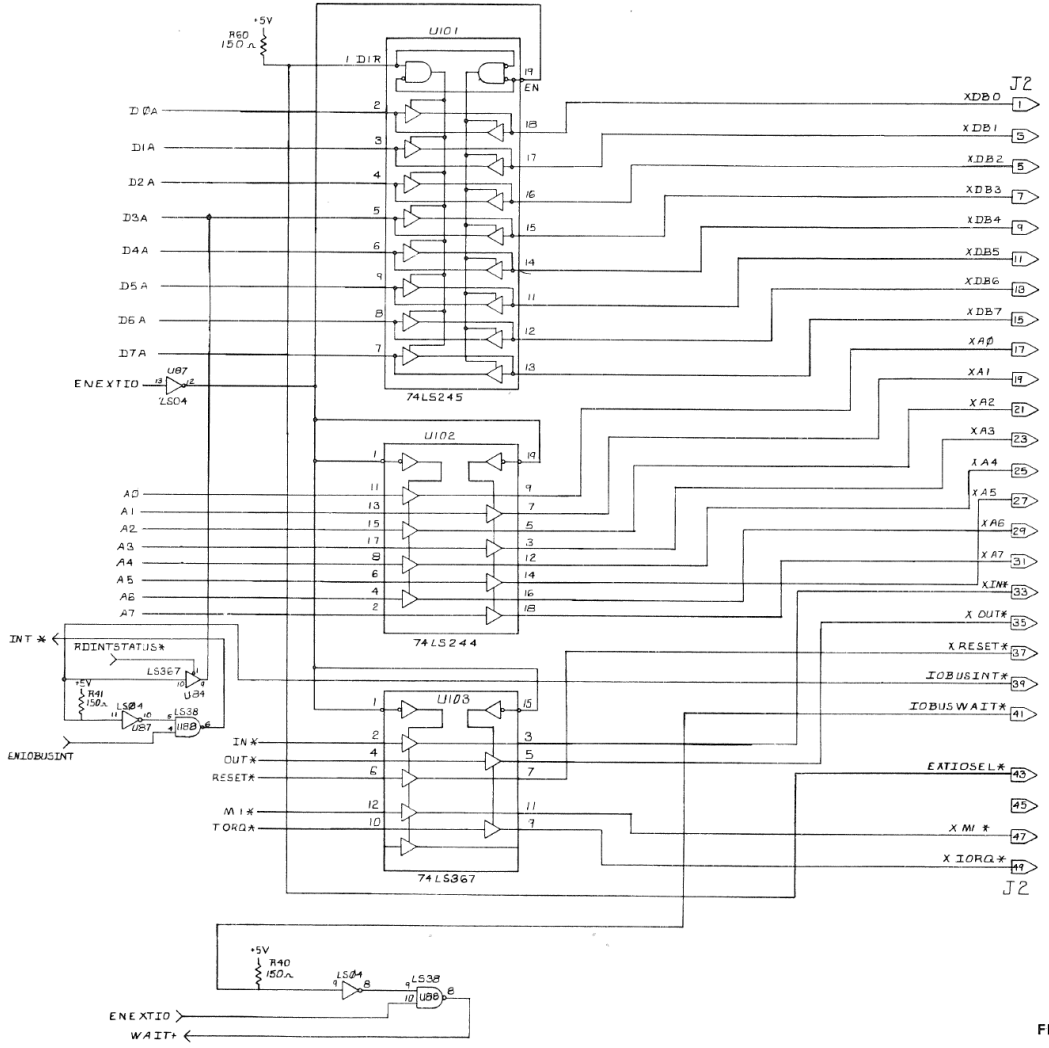
Service Manual

26-1061/1062/1063

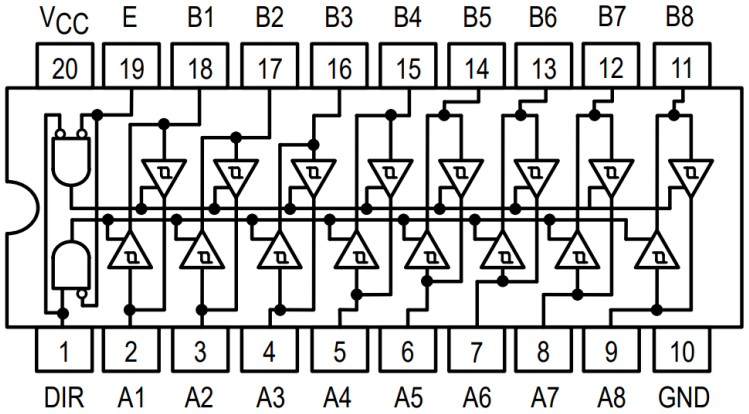
TRS-80®
MODEL III
MICROCOMPUTER

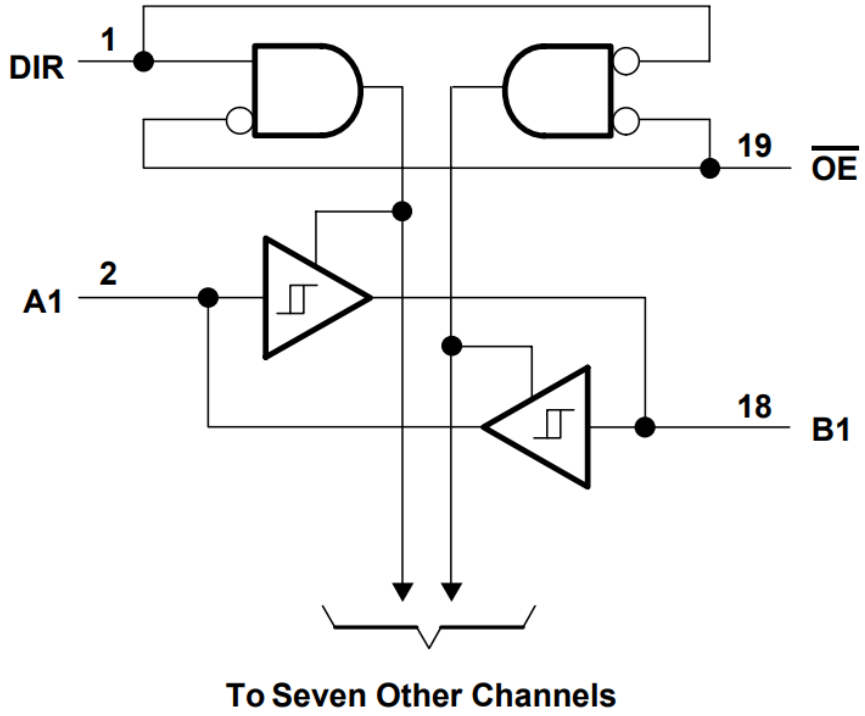
Catalog Numbers 26-1061/1062/1063

CUSTOM MANUFACTURED IN U.S.A. BY RADIO SHACK A DIVISION OF TANDY CORPORATION


+5V
R60
150Ω
U101
1 DIR
19 EN
D0A
D1A
D2A
D3A
D4A
D5A
D6A
D7A
2
3
4
5
6
9
8
7
18
17
16
15
14
11
12
13
74LS245
U87
ENEXTIO
LS04
J2
XDB0 1
XDB1 5
XDB2 5
XDB3 7
XDB4 9
XDB5 11
XDB6 13
XDB7 15
XA0 17
XA1 19
XA2 21
XA3 23
XA4 25
XA5 27
XA6 29
XA7 31
XINT* 33
XOUT* 35
XRESET* 37
IOBUSINT* 39
IOBUSWAIT* 41
EXTIOSEL* 43
45
XMI* 47
XIORQ* 49
J2
U102
A0
A1
A2
A3
A4
A5
A6
A7
1
11
13
15
17
8
6
4
2
19
9
7
5
3
12
14
16
18
74LS244
INT*
RDINTSTATUS*
LS367
U89
+5V
R91
150Ω
LS04
U87
LS38
U88
ENIOBUSINT
U103
IN*
OUT*
RESET*
MI*
IORQ*
1
2
4
6
12
10
15
3
5
7
11
9
74LS367
+5V
R90
150Ω
LS04
LS38
U88
ENEXTIO
WAIT*

SN74LS245



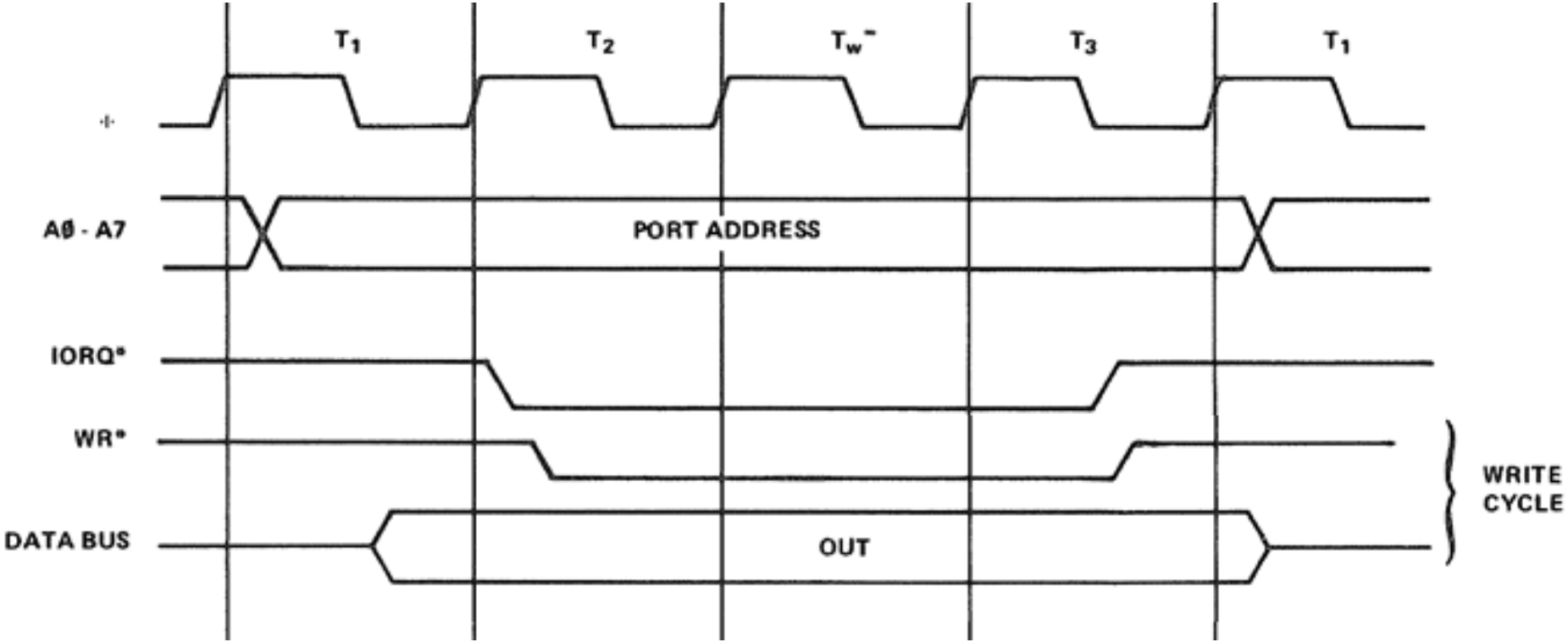


- Octal bus transceiver
- 3-state outputs

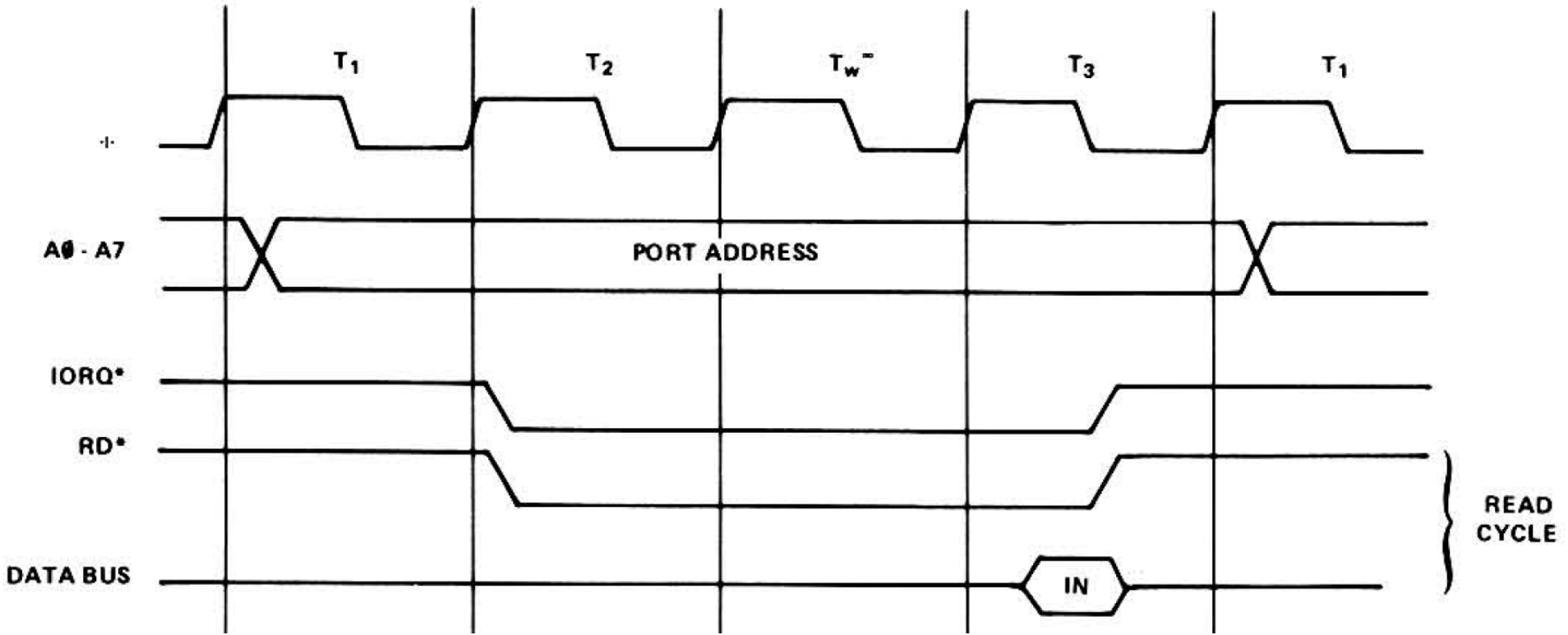
I/O Bus Pins

Name	Description	Pins
D[0-7]	Data lines	1,3,…,15
A[0-7]	Address lines	17,19,…,31
RD*	Read in progress	33
WR*	Write in progress	35
IORQ*	I/O request	49
EXTIOSEL*	Assert for read operation	43
IOBUS_WAIT*	Force wait-states on Z80	41
IOBUS_INT*	Signal interrupt to Z80	39

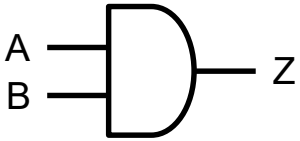
I/O Bus Write Cycle



I/O Bus Read Cycle

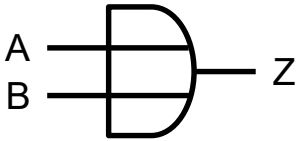


Boolean Logic



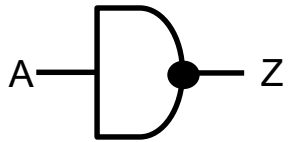
AND

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1



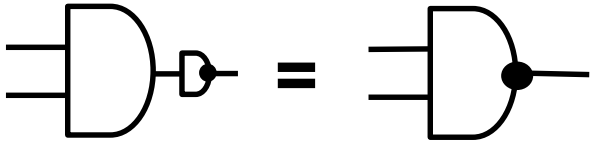
OR

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1



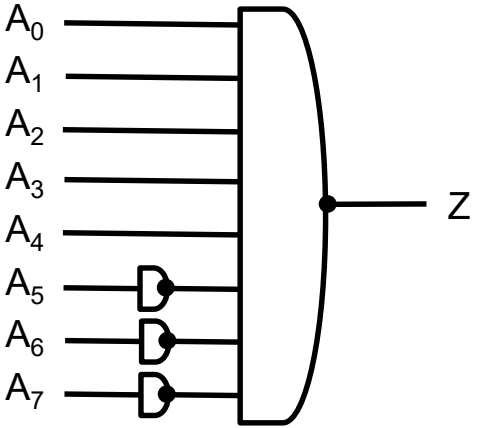
NOT

A	Z
0	1
1	0



NAND

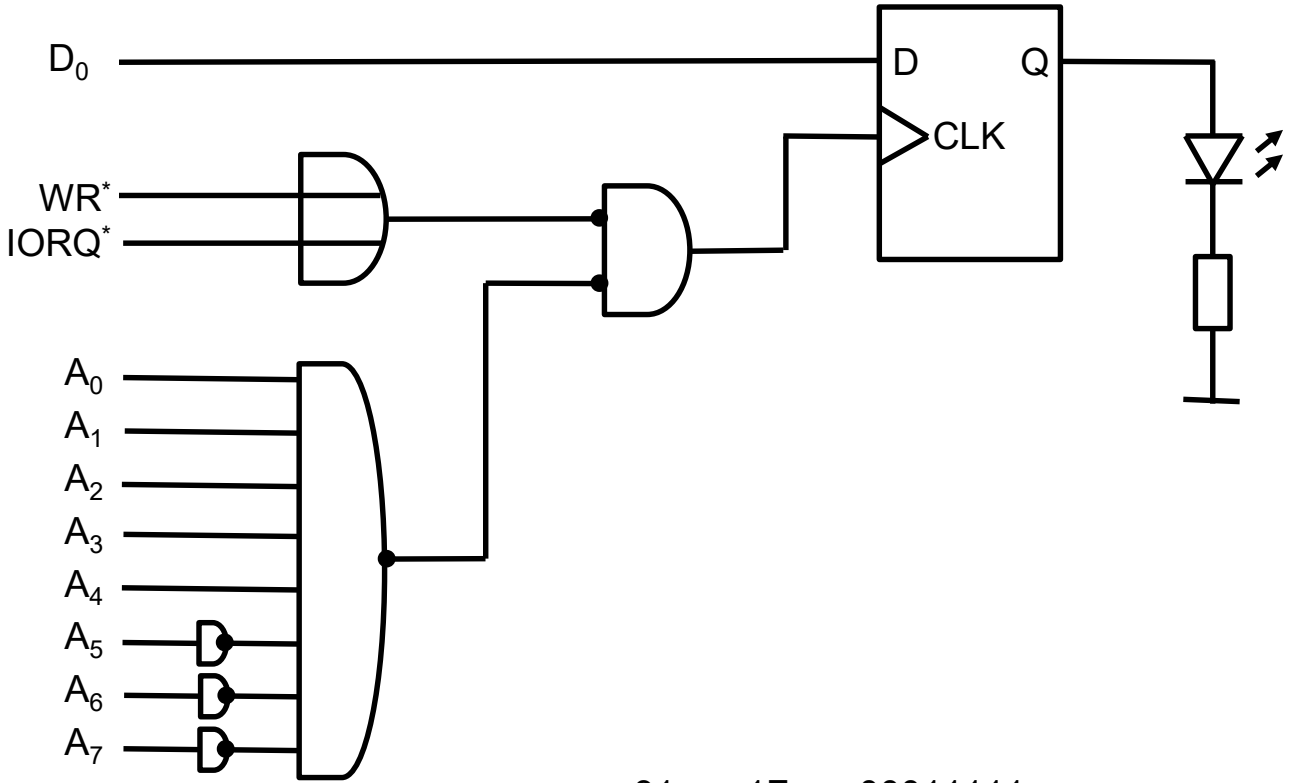
Boolean Logic Example



Z = 0 if and only if A_0 … A_4 = 1 and A_5 … A_7 = 0

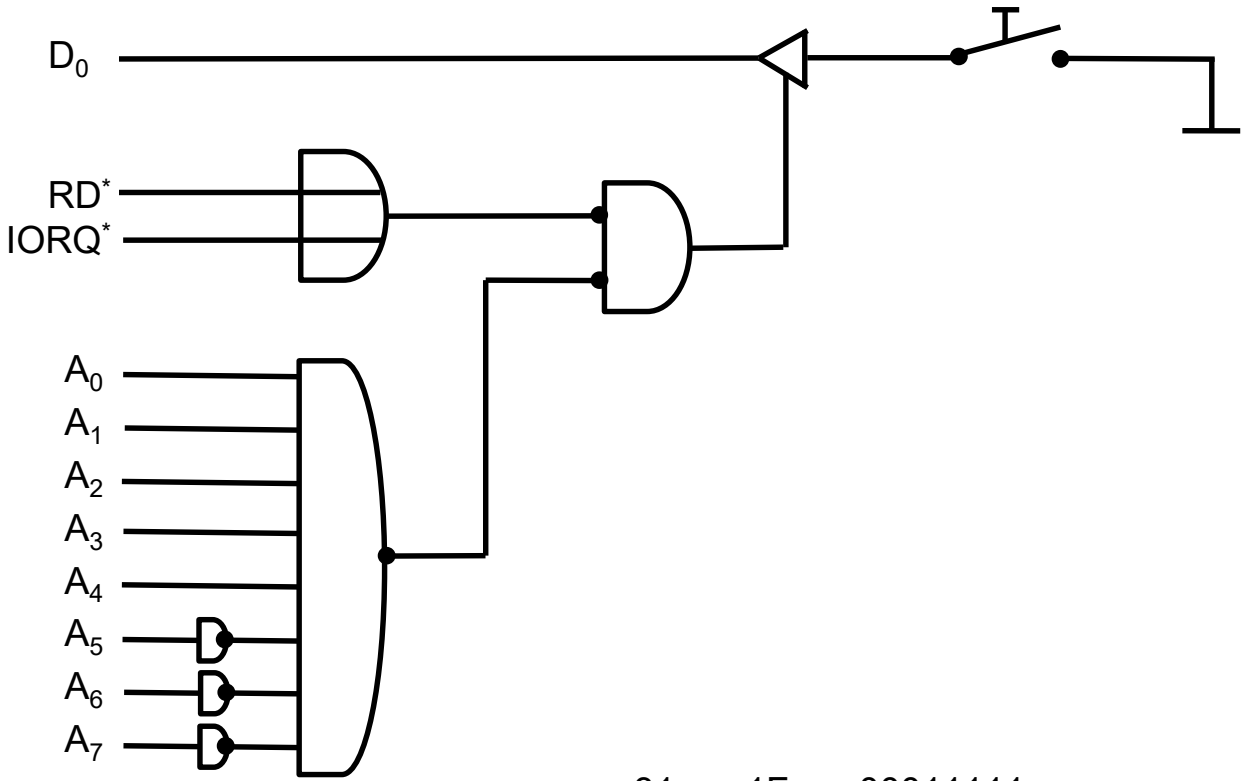
$31_{10} = 1F_{16} = 00011111_2$

1 Bit Output



$31_{10} = 1F_{16} = 00011111_2$

1 BitInput



$31_{10} = 1F_{16} = 00011111_2$

ATF16V8B

- Generic Array Logic (GAL)
- Ideal for prototyping and minimizing simple Boolean equations.
- Precursor of FPGAs.
- Can be erased and reprogrammed.
- Up to 16 Inputs.
- 8 I/O.
- Requires a programmer (e.g., TL866II)

TL866II Plus

GAL Pin Assignments

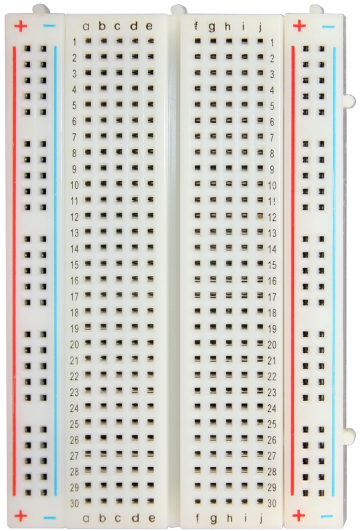
```
/* Inputs  */
Pin [2..9] = [A7..0];
Pin 17  =  !RD_N;
Pin 18  =  !WR_N;
Pin 19  =  !IORQ_N;

/* Outputs */
Pin 14  =  EXTIO_SEL;
Pin 13  =  !ARDUINO_SEL_N;
```

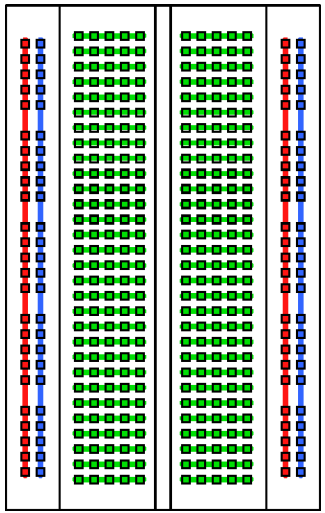
GAL Boolean Equations

```
/* Equations */
io_bus_sel = IORQ_N & (RD_N # WR_N);

/* Arduino is selected at port 31 */
arduino_sel = !A7 & !A6 & !A5 & A4 &
               A3 & A2 & A1 & A0 & io_bus_sel;
ARDUINO_SEL_N = arduino_sel;

/* EXTIO_SEL_N allows the TRS-80 to read */
EXTIO_SEL = arduino_sel & RD_N;
```

Arduino Starter Kit

Breadboards





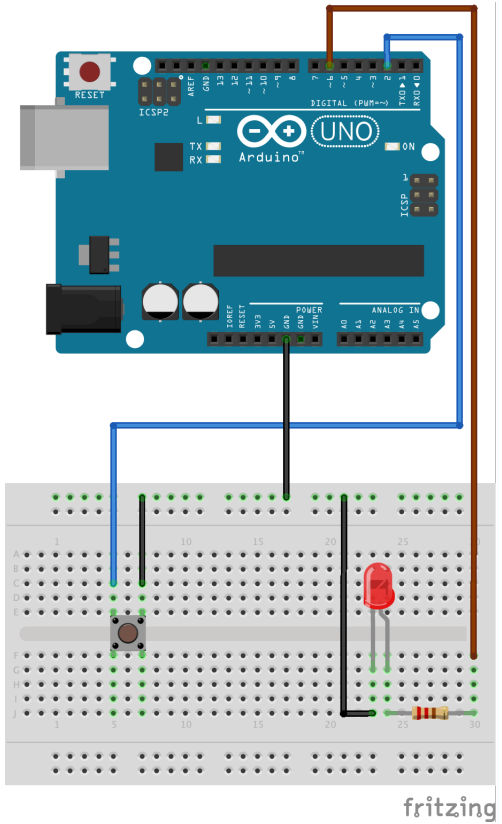
- Breadboards are used for fast prototyping.
- Allows to build an electronic circuit without soldering.
- Pins are connected as shown in the schematic on the right.

General Purpose Input/Output (GPIO)

- Generic pin with no pre-defined purpose.
- Can be configured either as input or output.
- Programmable interface:
  - Read state of a binary device (e.g., push button).
  - Control on/off state of a binary output device (e.g., LED)
- Arduino Uno:
  - 14 digital GPIO pins.

“Hello World” Arduino

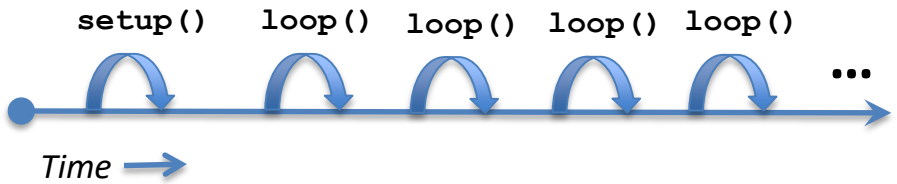
- First circuit that makes use of the Arduino.
- Push button is connected to GPIO pin 2.
- LED is connected to GPIO pin 6.
- Both LED and the push button are also connected to ground to form a complete circuit with their respective GPIO pins.
- Pushing the button will not turn on the LED.
- Arduino needs to be programmed to implement desired behavior.



Arduino Sketches

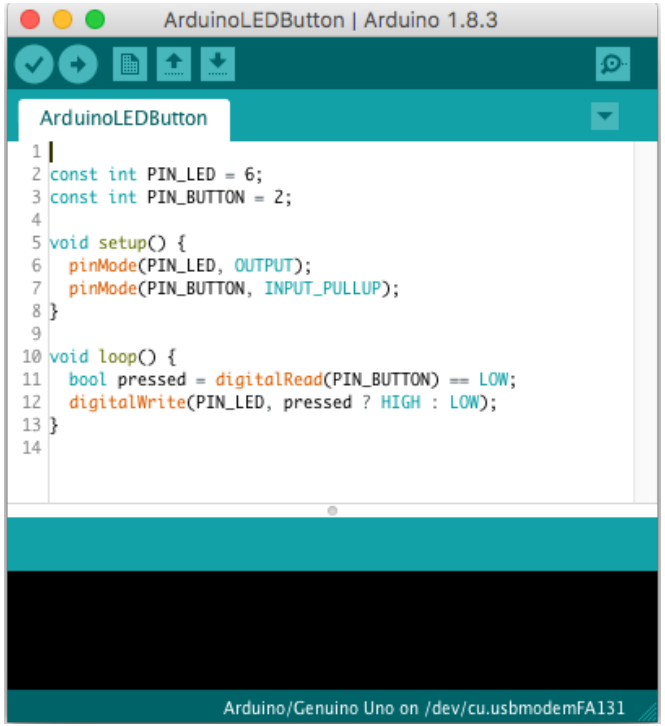
- Programs for the Arduino are called **Sketches**.
- Sketches can be written in C/C++.
- No operating system!
- Only one thread of execution.
- Support for common POSIX functions (e.g., **`malloc()`**, **`strlen()`**, etc)
- IDE automatically includes common header files.
- Popular IDEs:
  - Arduino IDE.
  - Atom with PlatformIO.

```
// Basic template for
// a sketch

void setup() {
  // ...
}

void loop() {
 // ...
}
```



Arduino IDE

- Download free Arduino IDE: https://www.arduino.cc/en/Main/Software
- Configuration:
  - *Tools > Board > Arduino/Genuino Uno*
  - *Tools > Programmer > AVR ISP*
  - *Tools > Port > /dev/XXX*
- Note: the Arduino needs to be connected to the laptop in order for it to show up under *Tools > Port*.



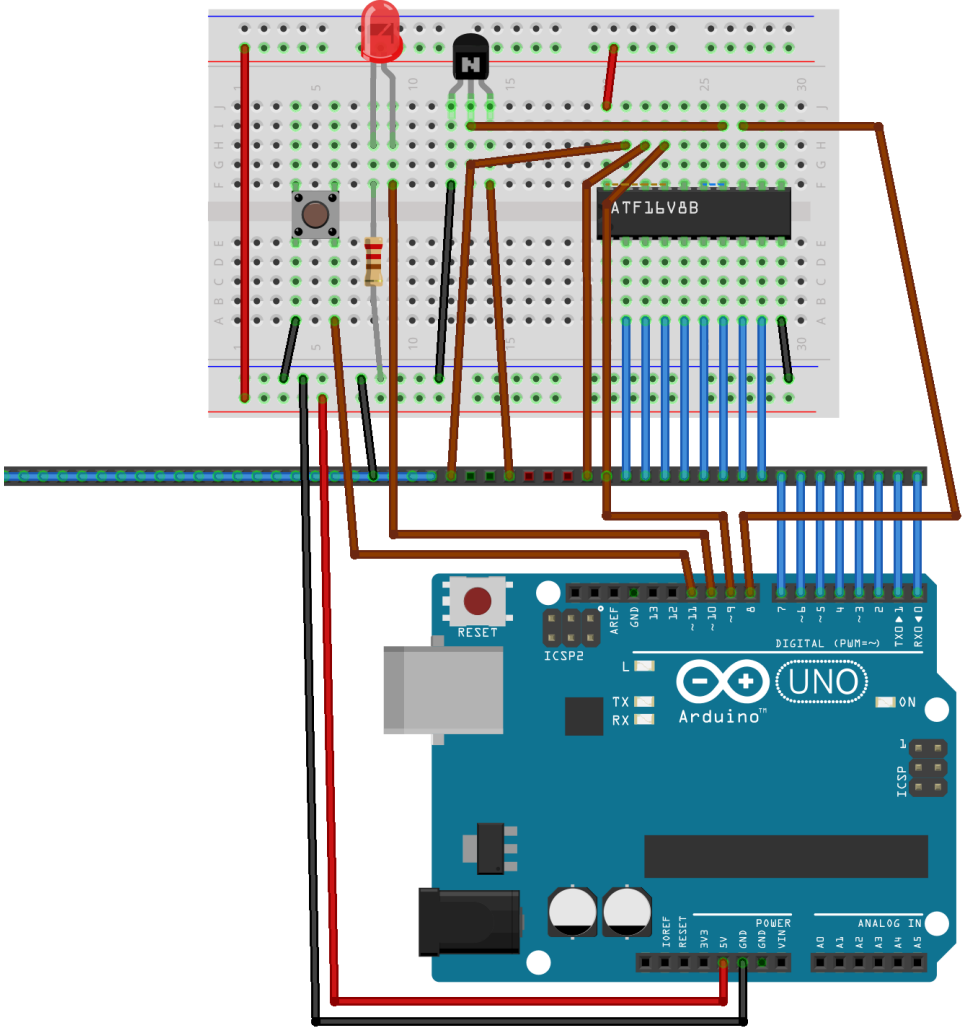
Sketch for Button/LED Circuit

```
const int LED_PIN = 6;
const int BUTTON_PIN = 2;

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  bool pressed = digitalRead(BUTTON_PIN) == LOW;
  digitalWrite(LED_PIN, pressed ? HIGH : LOW);
}
```

- This sketch assumes the wiring as shown on the “Hello World” for Arduino slide.
- GPIO pins can be configured either as input (**`INPUT_PULLUP`**) or output (**`OUTPUT`**) via **`pinMode()`**
- **`digitalRead()`** can return either **`HIGH`** or **`LOW`**. For a pin configured as **`INPUT_PULLUP`** it will return **`LOW`** when the button is pressed. It will also activate an internal pull-up resistor on that pin.
- The value of **`HIGH`** or **`LOW`** can be set for an output pin via **`digitalWrite()`**
- This sketch will constantly update the LED with each call to **`loop()`**

Fritzing

- Design tool fritzing.org allows simple schematics.
- Uses ATF16V8B, transistor, LED, resistor, button.
- Button: pseudo input.
- LED: pseudo output.



Main Polling Loop

```
#define PIN8 (1 << 0)
#define PIN9 (1 << 1)

void setup() {
  setup_app();
  pinMode(8, INPUT);
  pinMode(9, INPUT);
  // Configure pins
  // 0-7 as input
  DDRD = 0;
  noInterrupts();
}
```

```
void loop() {
  // Wait for ARDUINO_SEL_N to go low
  while (PINB & PIN8) ;
  if (PINB & PIN9) {
    // Read from I/O bus
    z80_out(PIND);
  } else {
    // Write to I/O bus
    DDRD = 0xff;
    PORTD = z80_in();
  }
  // Wait for ARDUINO_SEL_N to go high
  while (!(PINB & PIN8)) ;
  DDRD = 0;
}
```

Examples

Blinking LED

```
10 OUT 236,16
20 OUT 31,0
30 FOR X=1TO1000:NEXT
40 OUT 31,1
50 FOR X=1TO1000:NEXT
60 GOTO 20
```

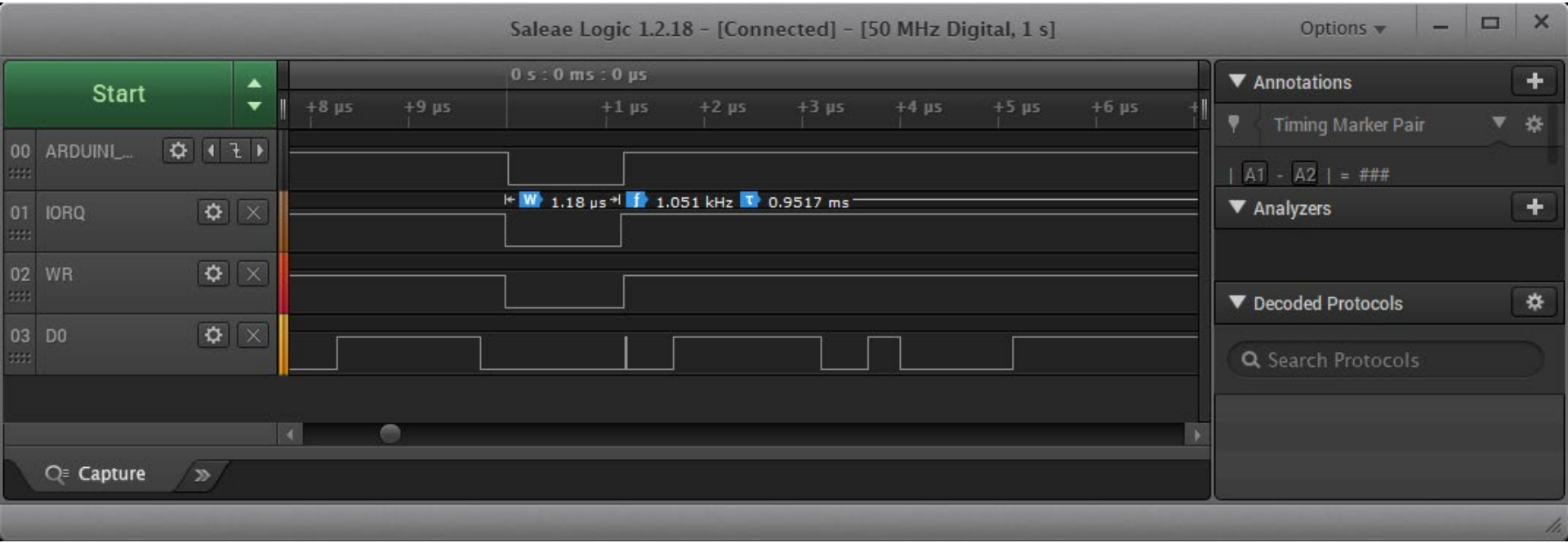
Pushing Button turns off LED

```
10 OUT 236,16
20 OUT 31,ABS(INP(31)-1)
30 GOTO 20
```

Read Button State

```
10 OUT 236,16
20 IF INP(31) = 1 THEN PRINT "BUTTON PRESSED"
30 GOTO 20
```

Logic Analyzer

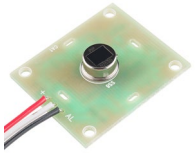


https://www.saleae.com/

Sensors

Humidity Sensor
HIH-4030

Atmospheric Sensor
BME280



Passive Infrared
Motion Sensor

Ultrasonic Sensor
HC-SR04

Soil Moisture

Sound Detector

Accelerometer, Gyroscope
Temperature – MPU6050

Photocell

Reed Switch
TMP36

Actuators/Displays

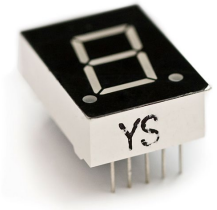
Gear motor
with Wheel Set

Solenoid

Water Valve

Relay

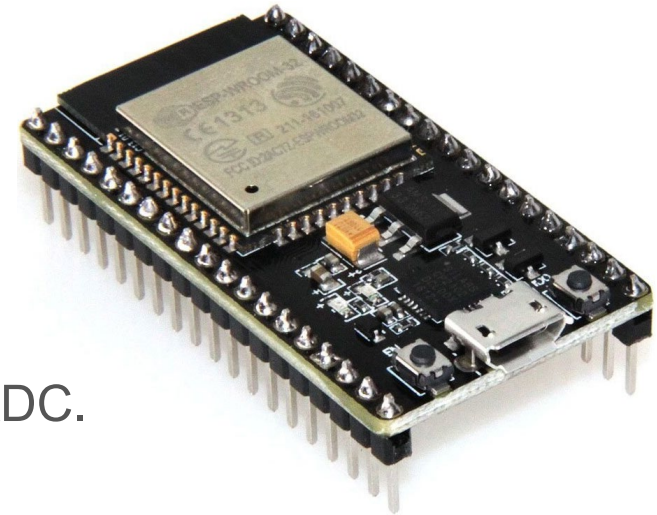
Liquid Crystal Display (LCD)



7 Segment Display

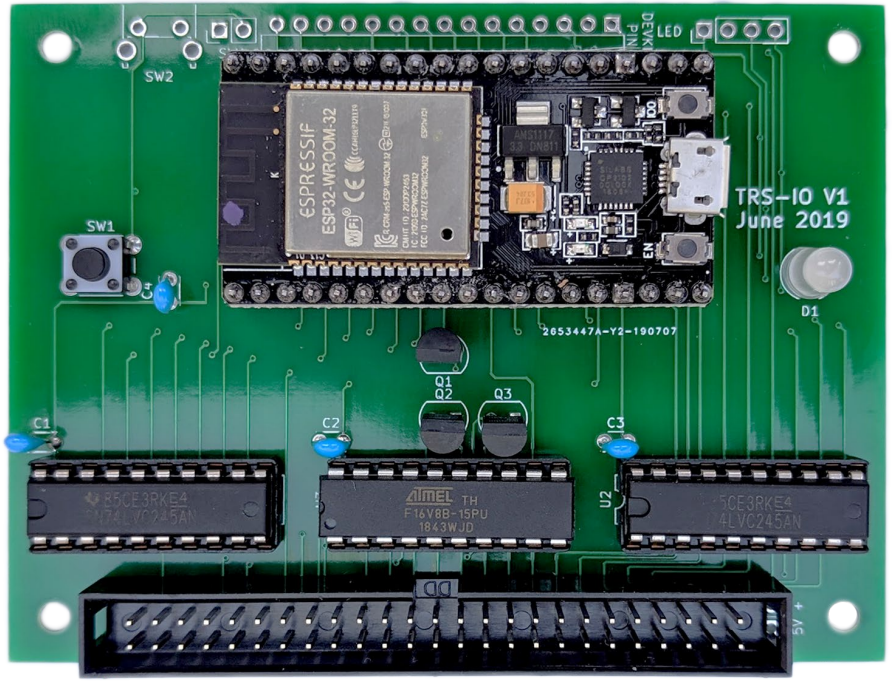
ESP-WROOM-32

- Low-cost microcontroller including WiFi and Bluetooth.
- Produced by Shanghai-based Chinese manufacturer, Espressif Systems.
- CPU: dual core 32-bit RISC Tensilica Xtensa LX6 @ 240 MHz.
- FreeRTOS
- Flash memory: 4MB
- SRAM: 520 kB
- WiFi:
  - IEEE 802.11 b/g/n
  - WEP or WPA/WPA2 authentication, or open networks.
- Interface: 16 GPIO pins, SPI, I2C, UART, 16-bit ADC.
- Price point: $11

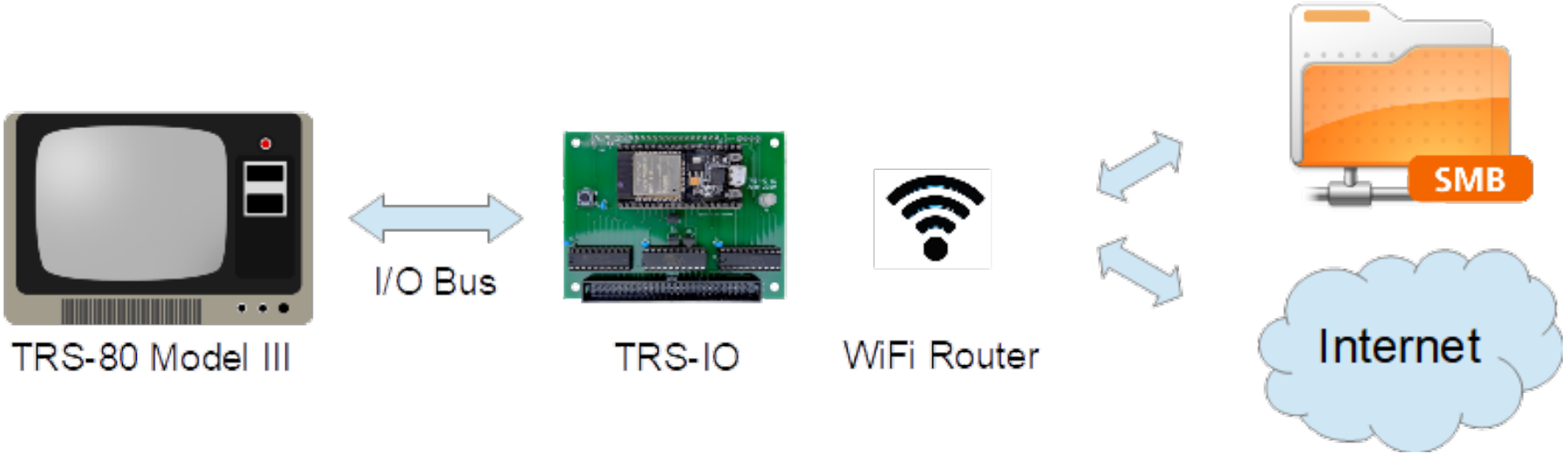


Introducing TRS-IO

- Same components as the RetroStoreCard (backward compatible).
- Only differences:
  - A0-A3 are routed to the ESP
  - GAL scans for port 31 and ports 0xC0-0xCF
- Features:
  - RetroStore access
  - trs-nic (Berkeley Socket API)
  - trs-fs (POSIX file access API)
  - FreHD
  - OTA



TRS-IO



All Good Things...

- Summary:
  - Old computers were great for learning.
  - Today's devices are tinker resistant.
  - Arduino to the rescue!
  - Connect an Arduino to the TRS-80's I/O Bus.
- Outlook:
  - TRS-IO: Connecting the TRS-80 to the World!
  - Display at Tandy Assembly's Exhibition.
- Complete source and slides freely available at: https://github.com/apuder/TRS-80-Arduino

Contact: arno@retrostore.org