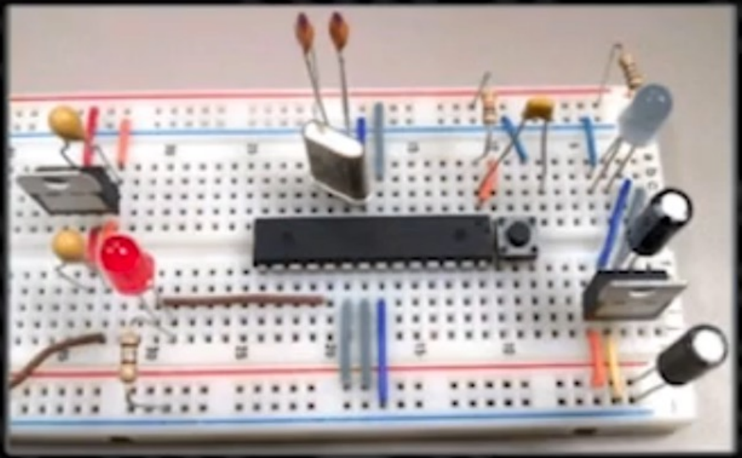
Hardware and Software

Hardware and Software

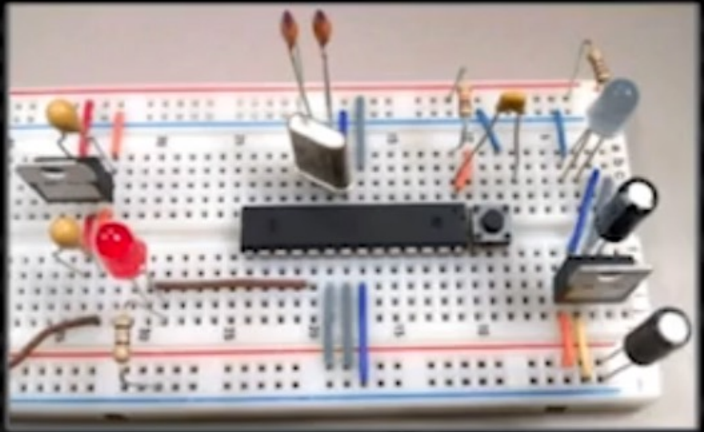
IoT devices are a combination of **hardware** and **software**



Hardware and software components must be designed together

Hardware and Software

IoT devices are a combination of **hardware** and **software**



```
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Hardware and software components must be designed together

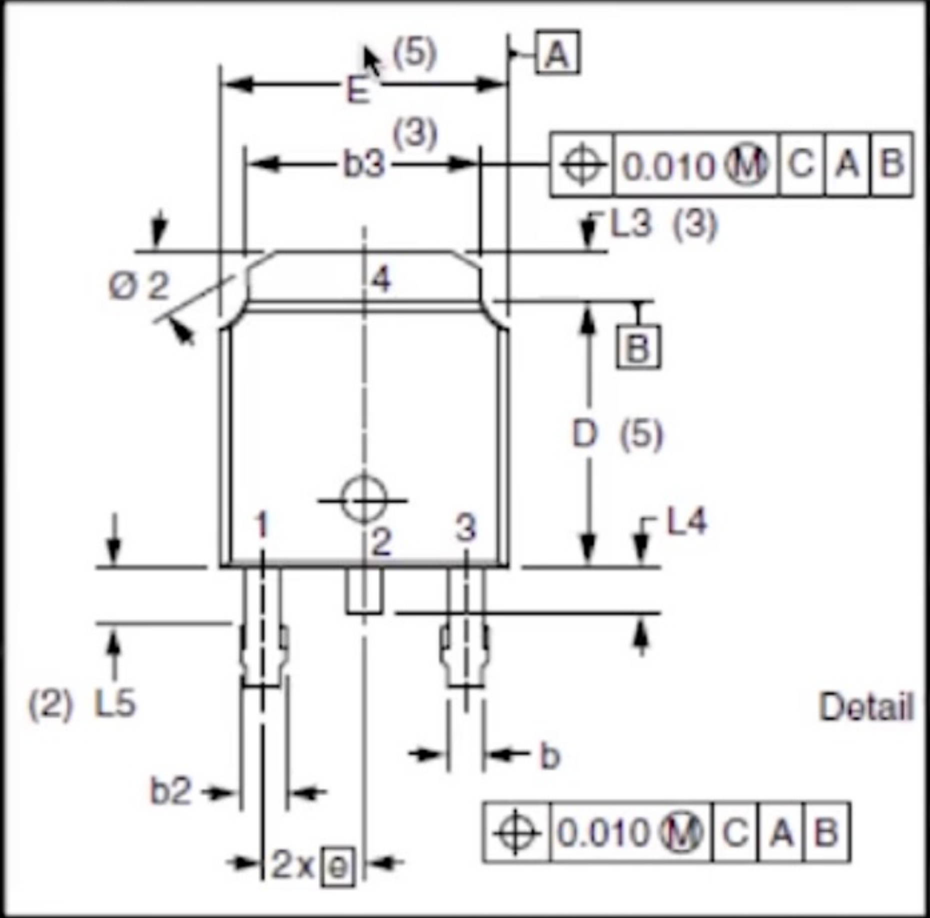
Datasheets

Datasheets give details of each hardware component

- Physical size
- Input/output pins
- Electrical parameters (max current, etc.)

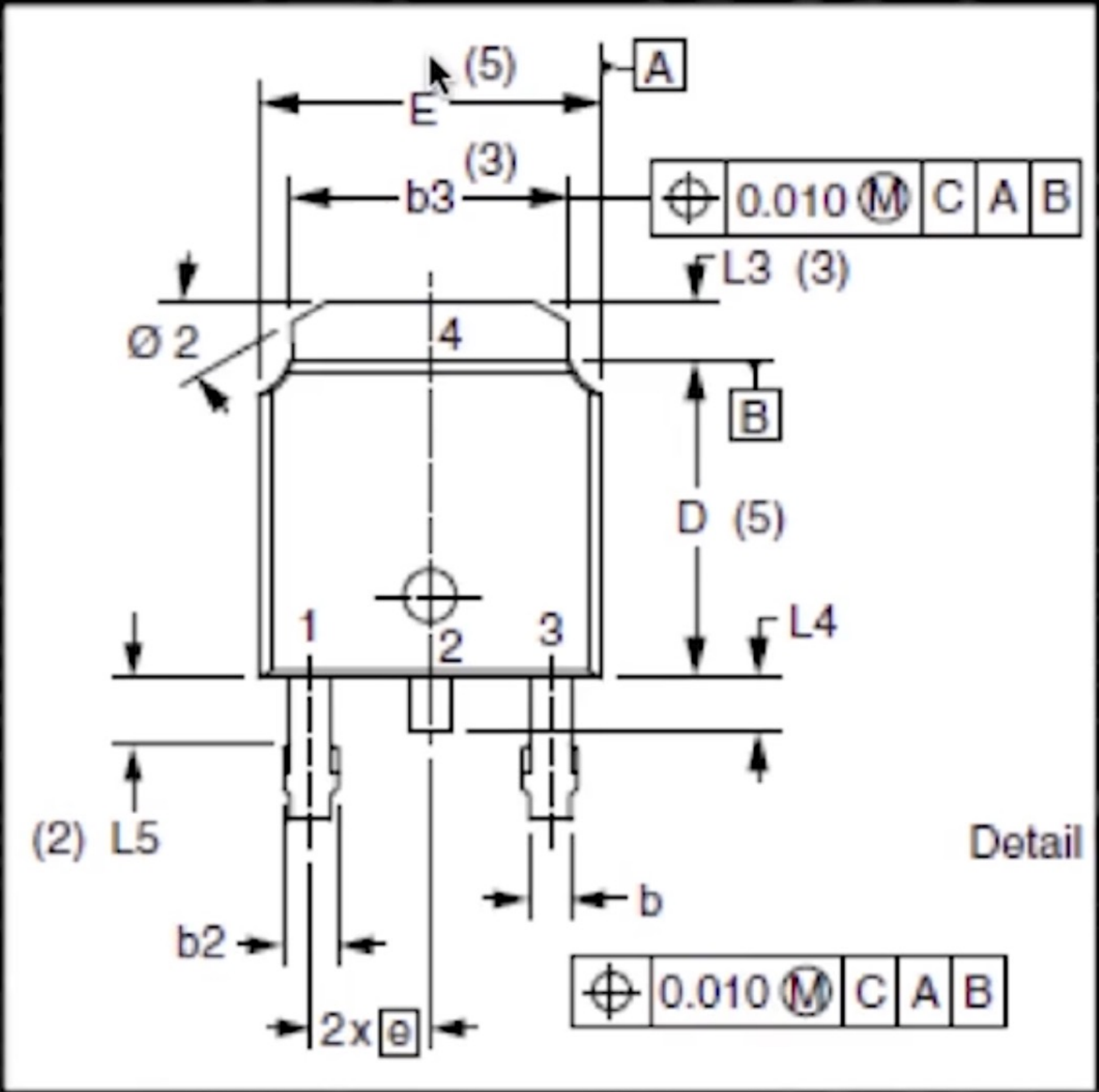
Datasheet Example

Schottky Rectifier



Size specifications for manufacturing

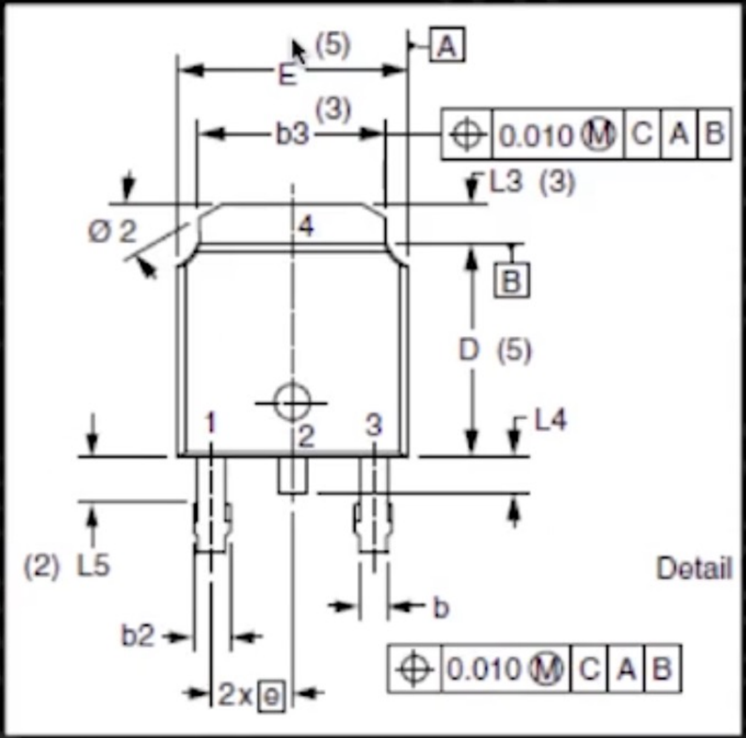
Schottky Rectifier



Size specifications for manufacturing

Datasheet Example

Schottky Rectifier



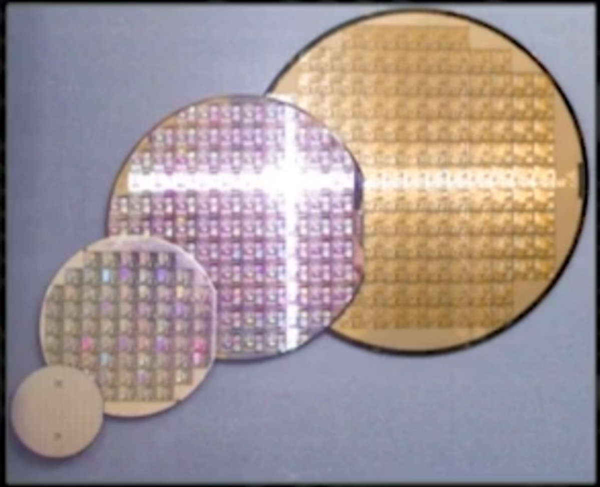
Size specifications for manufacturing

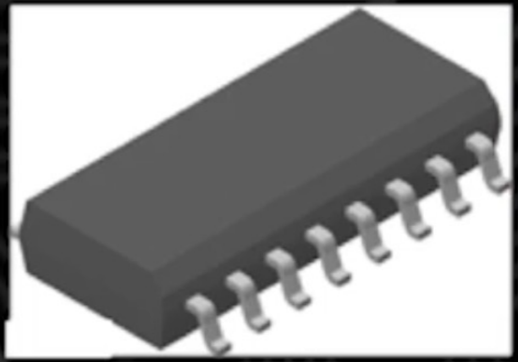
Electrical and thermal parameters

ELECTRICAL SPECIFICATIONS					
PARAMETER	SYMBOL	TEST CONDITIONS		VALUES	UNITS
Maximum forward voltage drop per leg See fig. 1	V_{FM} (1)	6 A	T_J = 25 °C	0.80	V
		12 A		0.95	
		6 A	T_J = 125 °C	0.65	
		12 A		0.78	

Integrated Circuits (ICs)

- A small electronic device made of a semiconductor material
  - Silicon, Galium Arsenide, etc.
- Chip is protected by a package
- Package has a set of pins to allow chip access





TSMC

From Wikipedia, the free encyclopedia

"Tai Ji Dian" redirects here. For the Hall in the Forbidden City, see Hall of the Supreme Principle.

Taiwan Semiconductor Manufacturing Company, Limited (**TSMC**; also called **Taiwan Semiconductor**)[4][5] is a Taiwanese multinational semiconductor contract manufacturing and design company. It is the world's most valuable semiconductor company,[6] the world's largest dedicated independent (pure-play) semiconductor foundry,[7] and one of Taiwan's largest companies,[8][9] with its headquarters and main operations located in the Hsinchu Science Park in Hsinchu. It is majority owned by foreign investors.[10]

Founded in Taiwan in 1987 by Morris Chang, TSMC was the world's first dedicated semiconductor foundry and has long been the leading company in its field.[11][12] When Chang retired in 2018, after 31 years of TSMC leadership, Mark Liu became Chairman and C. C. Wei became Chief Executive.[13][14] It has been listed on the Taiwan Stock Exchange (TWSE: 2330) since 1993; in 1997 it became the first Taiwanese company to be listed on the New York Stock Exchange (NYSE: TSM). Since 1994, TSMC has had a compound annual growth rate (CAGR) of 17.4% in revenue and a CAGR of 16.1% in earnings.[15]

Most of the leading fabless semiconductor companies such as AMD, Apple, ARM, Broadcom, Marvell, MediaTek, and Nvidia, are customers of TSMC, as are emerging companies such as Allwinner Technology, HiSilicon, Spectra7, and Spreadtrum.[16] Leading programmable logic device companies Xilinx and previously Altera also make or made use of TSMC's foundry services.[17] Some integrated device manufacturers that have their own fabrication facilities, such as Intel, NXP, STMicroelectronics and Texas Instruments, outsource some of their production to TSMC.[18][19] At least one semiconductor company, LSI, re-sells TSMC wafers through its ASIC design services and design IP portfolio.

TSMC has a global capacity of about thirteen million 300 mm-equivalent wafers per year as of 2020, and makes chips for customers with process nodes from 2 microns to 5 nanometres. TSMC is the first foundry to provide 7-nanometre and 5-nanometre (used by the 2020 Apple A14 and M1 SoC) production capabilities, and the first to commercialize extreme ultraviolet (EUV) lithography technology in high volume.

Taiwan Semiconductor Manufacturing Company (TSMC) Ltd.

Native name	台灣積體電路製造股份有限公司
Type	Public
Traded as	TWSE: 2330 NYSE: TSM
ISIN	US8740391003
Industry	Semiconductors GPUs Graphics cards Consumer electronics Computer hardware
Founded	Industrial Technology Research Institute,

Microcontroller Characteristics

Datapath Bitwidth

- Number of bits in each register
- Determines accuracy and data throughput

Input/Output Pins

- Need enough pins to support your application

Performance

- Clock rates are slower than desktop

Other Microcontroller Features

Timers

- Needed for real-time applications

Analog-to-Digital Converters

- Used to read input from analog sensors

Low-power modes

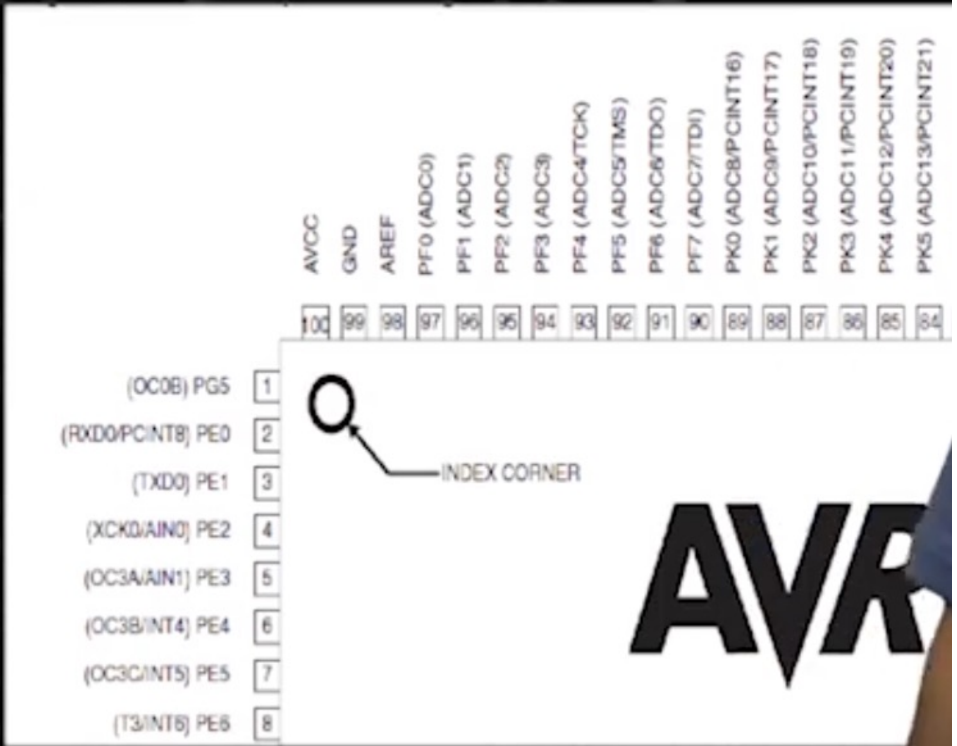
- Power saving is key

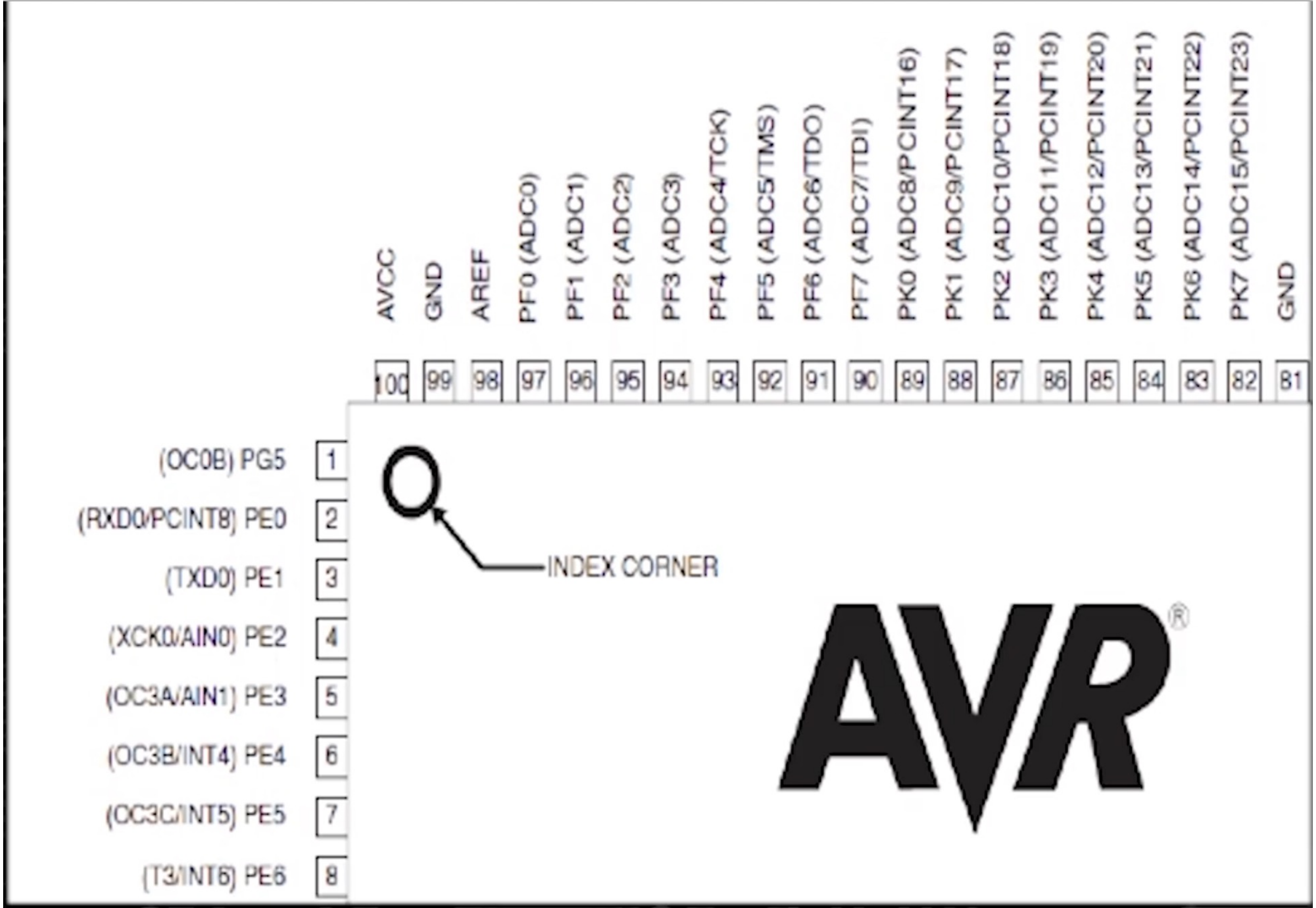
Communication protocol support

- Interface with other ICs
- UART, I2C, SPI, etc.

AVR ATmega2560

- 8-bit microcontroller
- Up to 16MHz
- 256KB of flash memory
- 4KB EEPROM, 8KB SRAM
- Peripherals




AVCC
GND
AREF
PF0 (ADC0)
PF1 (ADC1)
PF2 (ADC2)
PF3 (ADC3)
PF4 (ADC4/TCK)
PF5 (ADC5/TMS)
PF6 (ADC6/TDO)
PF7 (ADC7/TDI)
PK0 (ADC8/PCINT16)
PK1 (ADC9/PCINT17)
PK2 (ADC10/PCINT18)
PK3 (ADC11/PCINT19)
PK4 (ADC12/PCINT20)
PK5 (ADC13/PCINT21)
PK6 (ADC14/PCINT22)
PK7 (ADC15/PCINT23)
GND
100
99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
(OC0B) PG5
1
(RXD0/PCINT8) PE0
2
(TXD0) PE1
3
(XCK0/AIN0) PE2
4
(OC3A/AIN1) PE3
5
(OC3B/INT4) PE4
6
(OC3C/INT5) PE5
7
(T3/INT6) PE6
8
INDEX CORNER
AVR®

Microcontroller Components

- Storage elements
  - Data stored in different components
  - Speed/Cost (area) tradeoff
- Register: stores a single value
  - Like a single memory location
  - Extremely fast & expensive
- Several special-purpose registers, used internally
- General-purpose registers, used by the program

Storage Elements

- Register file: Stores many values
  - A set of registers; each has an address
  - Acts just like a memory
  - Extremely fast, expensive
- Instruction operands are here
  - `add $r3, $r2, $r1`
- Can only read one or two registers at a time
- May contain ~32 registers

Memories

- **Cache**: Stores many values
    - Slower than a register file
    - Cheaper than a register file
    - Still fairly fast and expensive
- Data cache holds data that the program operates on
- Instruction cache holds program instructions

Main Memory

- Very big: Gigabytes (Gb)
- Not in the CPU
- Connected to the CPU via **system bus**
- Memory access is slow

Von Neumann Bottleneck:

- Memory is much slower then the CPU

Software Translation

- Machine language: CPU instructions represented in binary
- Assembly language: CPU instructions with mnemonics
  - Easier to read
  - Like machine language
- High-level language: commonly used languages (C, C++, Java)
  - Easier to use

Compilation and Interpretation

- **Compilation:** translate instructions once before running the code
  - C, C++, Java (partially)
  - Translation occurs only once, saves time
- **Interpretation: translate instructions while code is executed**
  - Basic, Java (partially)
  - Translation occurs every execution
  - Translation can adapt to runtime situation

Python vs C/C++

- Python is interpreted -- a scripting language
- Python is easier to work with, if speed is not primary

- Python is interpreted -- a scripting language
- Python is easier to work with, if speed is not primary

VS

- C and C++ are compiled
- C and C++ are more common for performance

```
a = b + c;
```

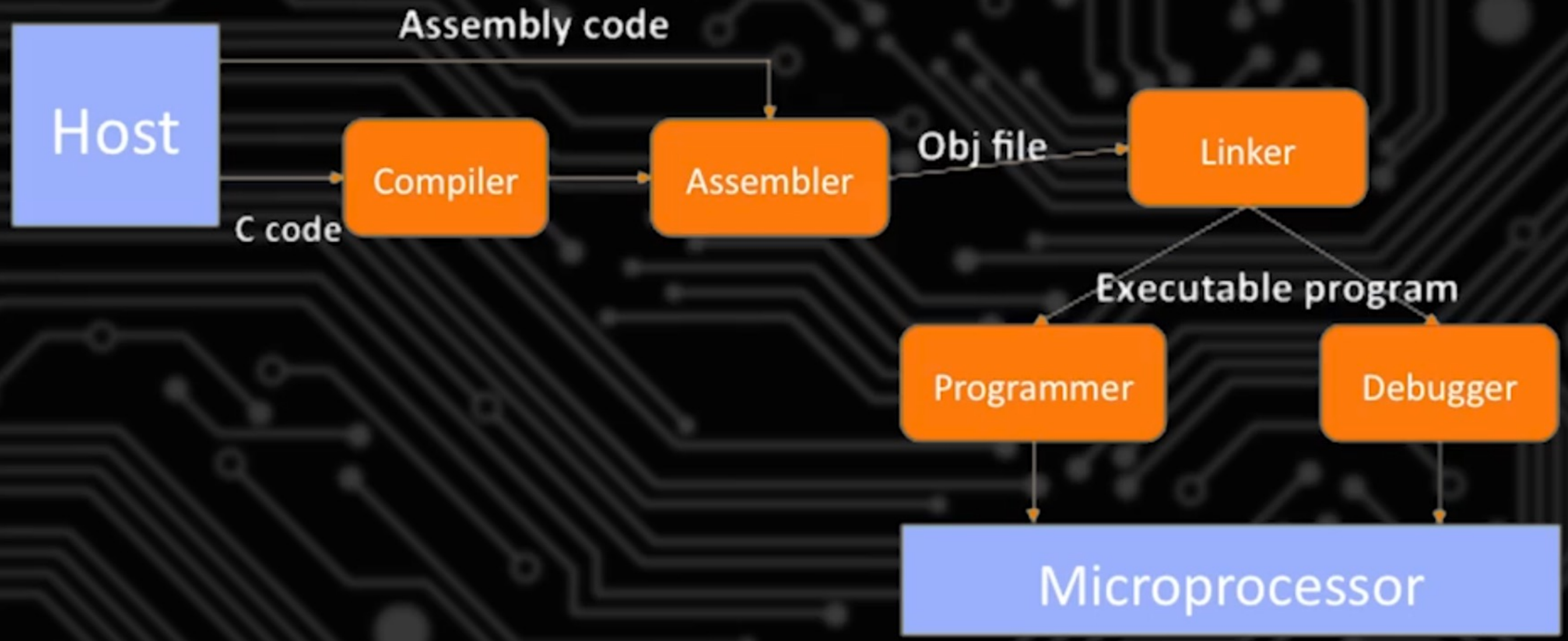
Compiler

`lw $r1, ($s1)`	**Load b from memory**
`lw $r2, ($s2)`	**Load c from memory**
`add $r3, $r2, $r1`	**Add b and c**
`sw $r3, ($s3)`	**Store result a in memory**

Assembler

10010001	00000011	00000010	00000001
add	**$r3**	**$r2**	**$r1**

Software Tool Chain



- Software written on a host
- Transferred to the flash memory in the microcontroller
- Tool chain is the set of tools needed to process the software

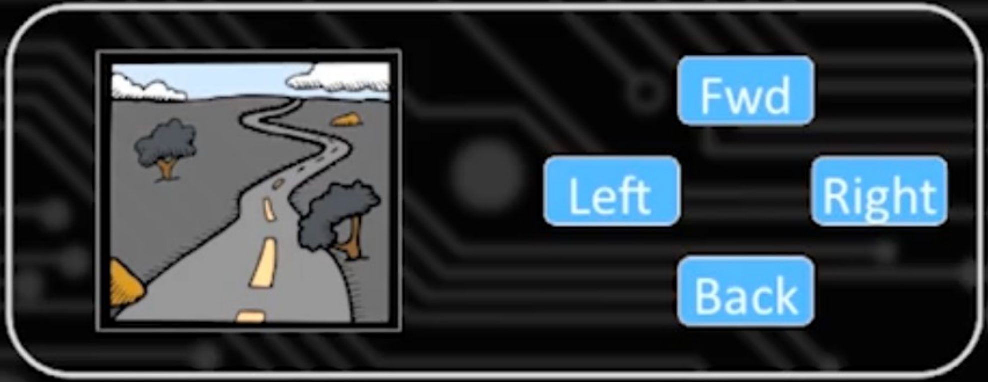
Operating System

- Manages other programs
  - Windows, Linux, IOS, etc.
- Allows many programs to be executed together
- Incorporates a nice user interface
- Needs processing power and memory
- Slows down the system, makes development easier

Operating System Example

Web-controlled car with a camera

- Car is controlled via the Internet
- Car has its own webserver (http://mycar/)
- Web interface allows user to control car and see camera images
- Car also has “auto brake” feature to avoid collisions



Web interface view

Multiple Tasks

- Assume that one microcontroller is being used
- At least four different tasks must be performed

1. **Send video data**: continuous while a user is connected
2. **Service motion buttons** : whenever button is pressed, may last seconds
3. **Detect obstacles**: continuous at all times
4. **Auto brake**: whenever obstacle is detected, may last seconds

Process/Task Support

- Main job of an OS is to support the **Process (Task) Abstraction**
- A **process** is an instantiation of a program; it must have access to:
    - the CPU
    - memory
    - other resources
        - I/O, ADC, timers, network, etc.
- OS must **manage resources**
    - Give processes fair access to the CPU
    - Give processes access to resources