

Introduction to the Microcontroller

John R. Wright, Jr., PhD, CSTM, CLSSGB, CSCE, F.ATMAE
AENG 467, Mobile Robotics
Millersville University

What is a Microcontroller?



<http://avrlab.net/atmel-atmega128-datasheet-download>

Brief History of the Microcontroller

- In 1971, the first microcontroller was invented by two engineers at Texas Instruments, according to the Smithsonian Institution.
 - Gary Boone and Michael Cochran created the TMS 1000, which was a 4-bit microcontroller with built-in ROM and RAM.
 - The same year that the microprocessor was invented at Intel
- The microcontroller was used internally at TI in its calculator products from 1972 until 1974, and was refined over the years.
- In 1974, TI offered the TMS 1000 for sale to the electronics industry. The TMS 1000 was available in various configurations of RAM and ROM sizes.

Brief History of the Microcontroller

- During the 1990s, microcontrollers with electrically erasable and programmable ROM (EEPROM) memories, such as flash memory, became available.
 - These microcontrollers could be programmed, erased and reprogrammed using only electrical signals.
 - Prior to the electrically reprogrammable devices, microcontrollers often required specialized programming and erasing hardware, which required that the device be removed from its circuit, slowing software development and making the effort more expensive.
 - With this limitation removed, microcontrollers were able to be programmed and reprogrammed while in a circuit so devices with microcontrollers could be upgraded with new software without having to be returned to the manufacturer. Many current microcontrollers, such as those available from Microchip and Atmel, incorporate flash memory technology.

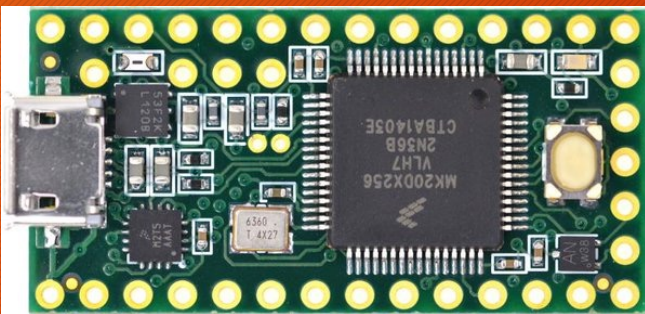
Some Popular Microcontrollers

BS2



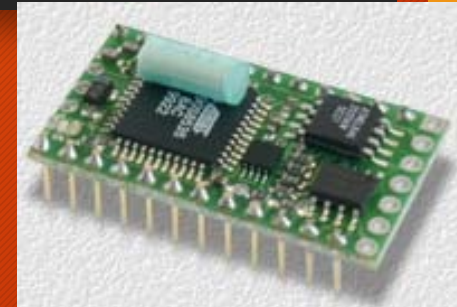
www.parallax.com

Teensy 3.2



<http://pattonrobotics.com/products/teensy-3-2>

BX-24p



www.basicx.com

Arduino Atmega32u4



www.radioshack.com

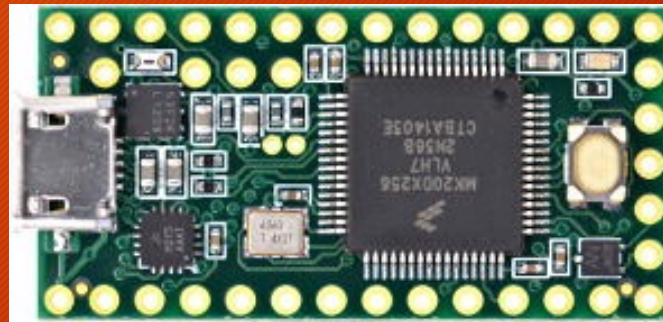
Feature	Teensy 3.0	Teensy 3.2 Teensy 3.1	Units
Price	19.00	19.80	US Dollars
Processor	MK20DX128VLH5	MK20DX256VLH7	
Core	Cortex-M4	Cortex-M4	
Rated Speed	48	72	MHz
Overclockable	96	96	MHz
Flash Memory	128	256	kbytes
Bandwidth	96	192	Mbytes/sec
Cache	32	256	Bytes
RAM	16	64	kbytes
EEPROM	2	2	kbytes
Direct Memory Access	4	16	Channels
Digital I/O	34	34	Pins
Voltage Output	3.3V	3.3V	Volts
Voltage Input	3.3V Only	5V Tolerant	Volts
Analog Input	14	21	Pins
Converters	1	2	
Resolution	16	16	Bits
Usable	13	13	Bits
Prog Gain Amp	0	2	
Touch Sensing	12	12	Pins
Comparators	2	3	
Analog Output	0	1	Pins
DAC Resolution	-	12	Bits

Timers	11 Total	12 Total	
FTM Type	2	3	
PWM Outputs	10	12	Pins
PDB Type	1	1	
CMT (infrared) Type	1	1	
LPTMR Type	1	1	
PIT (interval) Type	4	4	
Systick	1	1	
RTC (date/time) **	1	1	
Communication			
USB	1	1	
Serial	3	3	
With FIFOs	1	2	
High Res Baud	3	3	
Fast Clock	2	2	
SPI	1	1	
With FIFOs	1	1	
I2C	1	2	
CAN Bus	0	1	
I2S Audio	1	1	
FIFO Size	4	8	

<https://www.pjrc.com/teensy/teensy31.html>

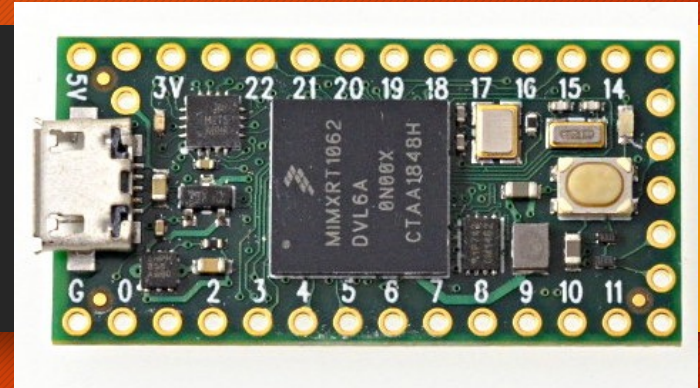
Teensy 3.2

- 32 bit ARM processor
 - ARM = Advanced RISC Machine
 - RISC = Reduced Instruction Set Computer
- 72MHz Cortex-M4 Technology (Can Overclock at 96MHz)
- 64K RAM

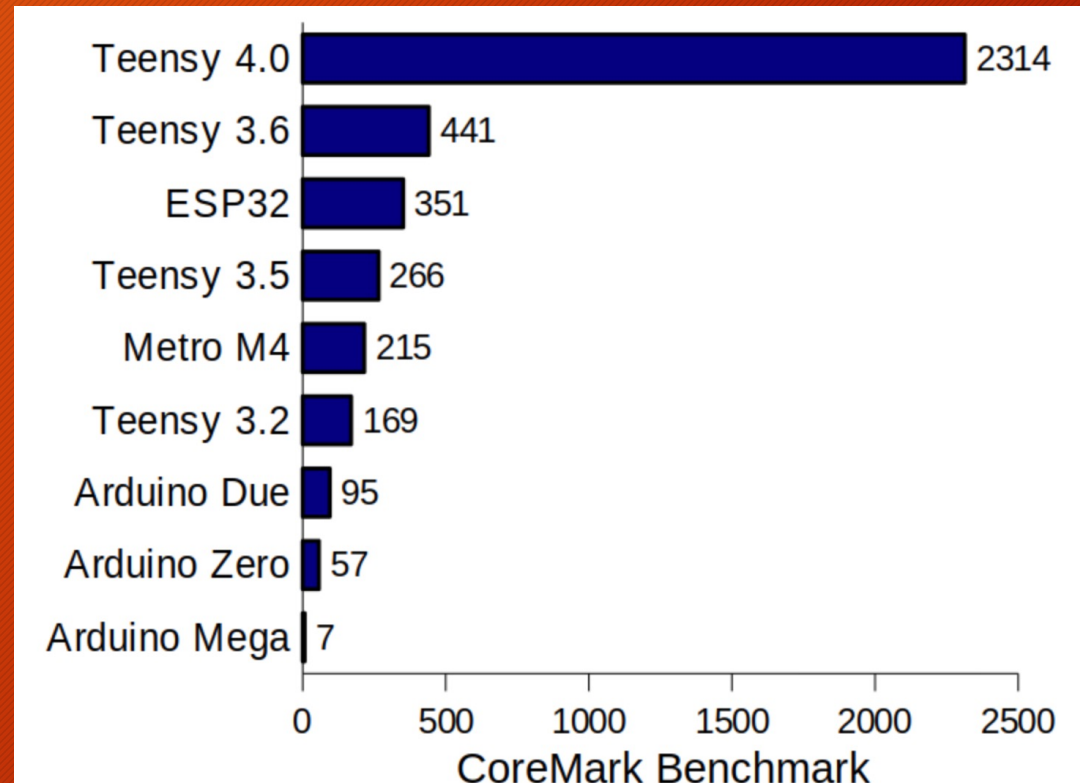


News Flash!

Introducing the Teensy 4.0 (~~Teensy 3.2~~ Everything Killer)



- 32 bit ARM processor
 - ARM = Advanced RISC Machine
 - RISC = Reduced Instruction Set Computer
- 600MHz Cortex-M7 Technology
- 1024K RAM
- HACKADAY ARTICLE 8/7/2019
 - <https://hackaday.com/2019/08/07/new-teensy-4-0-blows-away-benchmarks-implements-self-recovery-returns-to-smaller-form/>



<https://www.pjrc.com/store/teensy40.html>

What is an IDE?

- IDE = Integrated Development Environment

- An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software.
- Typically, an IDE contains a code editor, a compiler or interpreter and a debugger that the developer accesses through a single graphical user interface (GUI).
- An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications.

<http://searchsoftwarequality.techtarget.co/definition/integrated-development-environment>

- JAVA uses Eclipse as its IDE
- We use C++ to Program the Teensy
 - via a patch called “Teensyduino”
 - utilizing the Arduino IDE

<https://www.pjrc.com/teensy/teensyduino.html>

Teensy Vs. Arduino

- Comparison

- <https://www.youtube.com/watch?v=rklfsYRshRQ>

- Getting Setup with Teensy Software! (14:36)

FYI:

Brian is the inventor of the Teensy Motherboard (Carrier Board) that we use and used to be with RoboOdyssey - maker of the BX24p (BasicX) microcontroller.

- Please be very careful when using the microcontroller without the carrier board - pins might short out the device if they should touch metal (use a wood table or use on top of a piece of paper)! Or one may plug it into a bread/proto board.

Step 1 - Download Arduino IDE

Step 2 - Download Teensyduino

Follow instructions and make sure this is installed in same location as the Arduino software.

Step 3 - Open the Arduino program once and then close it.

Step 4 - Open Arduino software, connect microcontroller to USB of computer

Step 5 - Under Tools...Board Menu select the microcontroller you are using

Step 6 - Write a short program and try to connect. You may need to be patient the first time as it takes a little time the very first time we launch the Teensyduino.

Intro to ARM Technology!

- https://www.youtube.com/watch?v=_R06xdksYTI



Cortex-A

Highest performance
Optimized for rich
operating systems

[Learn more about the
Cortex-A series processors](#)



Cortex-R

Fast response
Optimized for high-
performance, hard real-
time applications

[Learn more about the
Cortex-R series processors](#)



Cortex-M

Smallest/lowest power
Optimized for discrete
processing and
microcontroller

[Learn more about the
Cortex-M series processors](#)



SecurCore

Tamper resistant
Optimized for security
applications

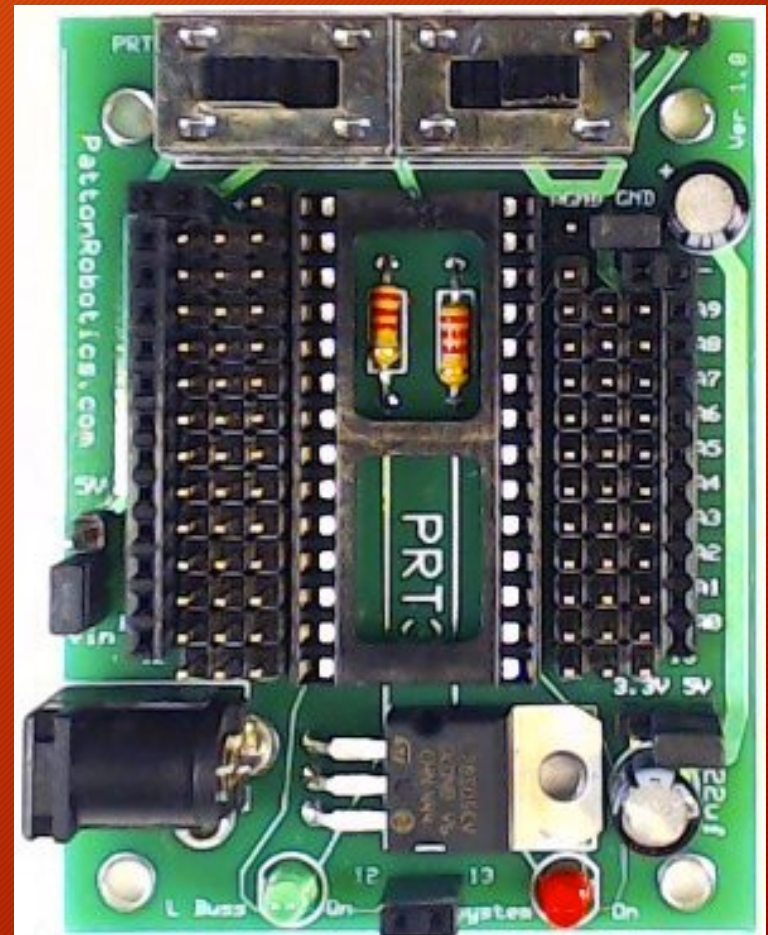
[Learn more about
SecurCore processors](#)

<https://www.arm.com/products/processors>

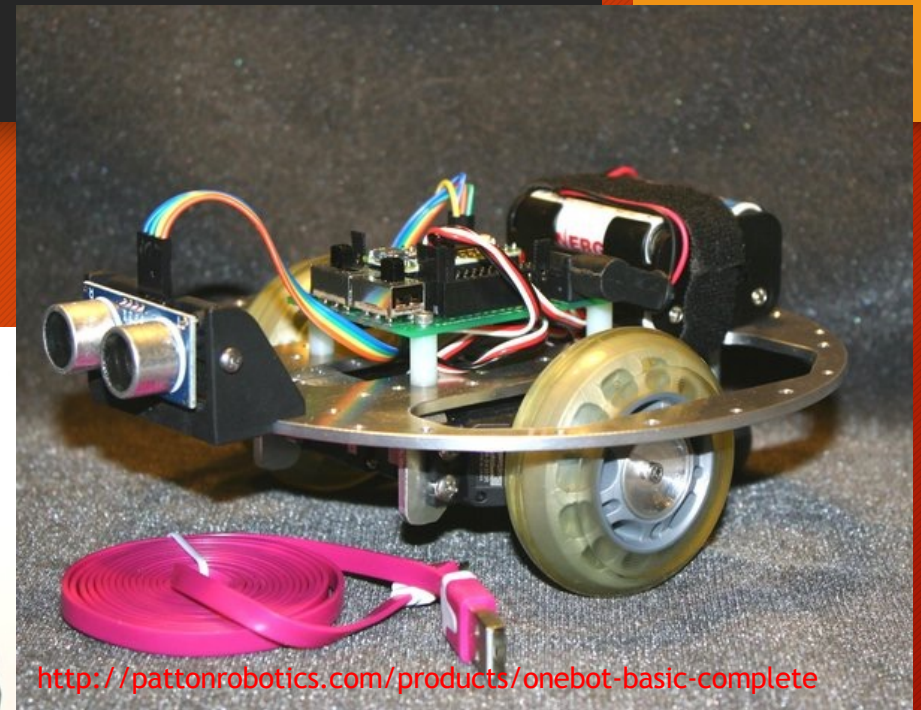
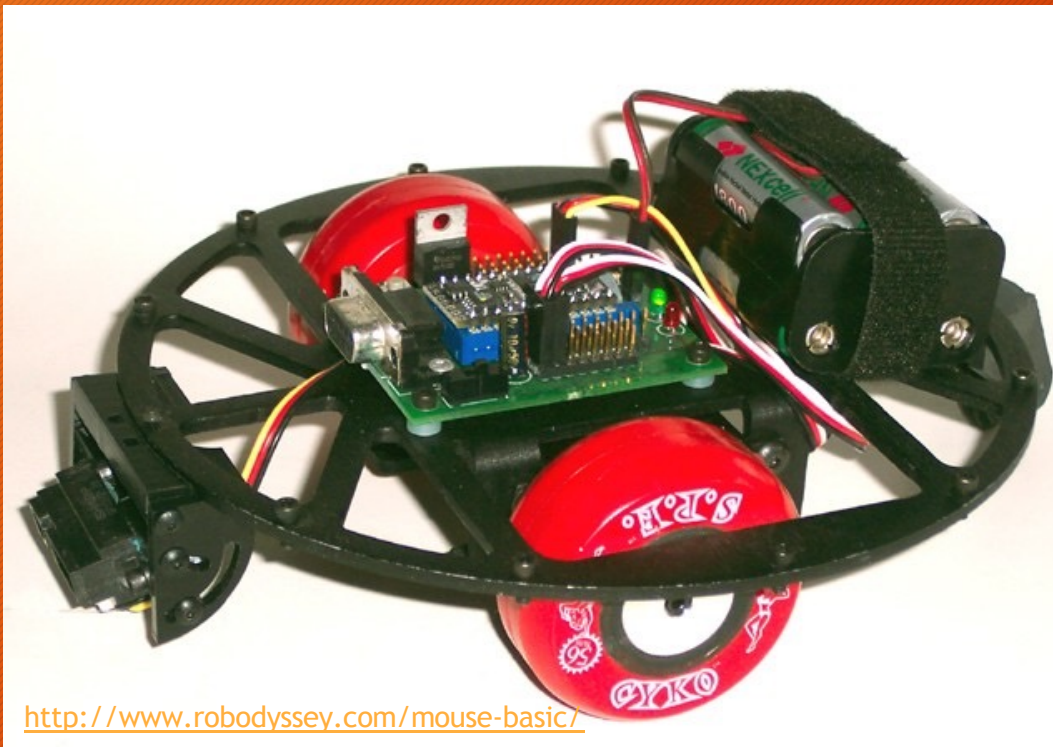
What is a Carrier Board?

PRT Motherboard tutorial 1 (12:08)
Intro to hardware and USB Voltage
Isolator

<https://www.youtube.com/watch?v=9in-VuT08g0&index=10&list=PLYI2Xb6BPCrriVQfwaxfCJdSIF6Chkh6R&t=73s>



Training Platform



Our R&D Application



Responsibilities (Senior Capstone R&D)

- To learn the code on your own using the provided resources. You will likely learn as you go - as you need the capability. There are multiple ways to approach the challenges and R&D project. Each developer may have a different approach so teaching will be done side-by-side as a result. You must research the code, however.
- You are responsible for the technology you check out! Replacement sensors and microcontrollers can be found here:
 - www.pattonrobotics.com
 - <https://www.pjrc.com/teensy/>
 - <https://openmv.io/products/openmv-cam-h7>
- You are expected to spend a considerable amount of time outside of class working on this R&D project. 75-100 hours of development time is typical and expected of a capstone experience.

I Code Every Night...

- All Night Longer - this semester!
- https://www.youtube.com/watch?v=z5_2xkOpVHU