

CS122A: Intermediate Embedded and Real Time Operating Systems

Jeffrey McDaniel

University of California, Riverside

Platforms

- ▶ Arduino
- ▶ Raspberry Pi
- ▶ ARM

Arduino



- ▶ Single-board microcontroller

Arduino



- ▶ Single-board microcontroller
- ▶ Provides all necessary circuitry for control tasks

Arduino



- ▶ Single-board microcontroller
- ▶ Provides all necessary circuitry for control tasks
- ▶ Inexpensive introduction to embedded system design

Arduino

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and running. The main text area contains the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 * This example code is in the public domain.  
 */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);             // wait for a second  
  digitalWrite(13, LOW);  // set the LED off  
  delay(1000);             // wait for a second  
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on IDE0014C01".

- ▶ Single-board microcontroller
- ▶ Provides all necessary circuitry for control tasks
- ▶ Inexpensive introduction to embedded system design
- ▶ Simple Integrated Development Environment (IDE)

Arduino

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and other IDE functions. The main text area displays the "Blink" sketch code, which is a standard example for controlling an LED. The code includes comments explaining the purpose and pin configuration. At the bottom, there is a status bar showing "1" and "Arduino Uno on /dev/ttyACM0".

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

- ▶ Single-board microcontroller
- ▶ Provides all necessary circuitry for control tasks
- ▶ Inexpensive introduction to embedded system design
- ▶ Simple Integrated Development Environment (IDE)
- ▶ Uses "**Wiring**" library to simplify coding

Raspberry Pi



- Fully functional computer

Raspberry Pi



- ▶ Fully functional computer
- ▶ More robust processing capabilities

Raspberry Pi



- ▶ Fully functional computer
- ▶ More robust processing capabilities
- ▶ Full Linux OS

Raspberry Pi



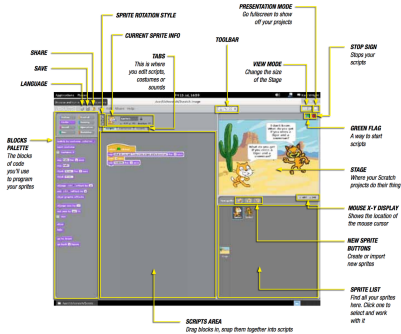
- ▶ Fully functional computer
- ▶ More robust processing capabilities
- ▶ Full Linux OS
- ▶ Supports multitasking

Raspberry Pi



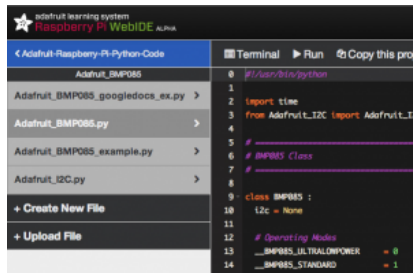
- ▶ Fully functional computer
- ▶ More robust processing capabilities
- ▶ Full Linux OS
- ▶ Supports multitasking
- ▶ requires more difficult hardware integration

Raspberry Pi



- Created to be used as teaching tools

Raspberry Pi



```
adafruit learning system
Raspberry Pi WebIDE ALPHA

< Adafruit-Raspberry-Pi-Python-Code
  Adafruit_BMP085
  Adafruit_BMP085_googleads_ex.py >
  Adafruit_BMP085.py >
  Adafruit_BMP085_example.py >
  Adafruit_I2C.py >
  + Create New File
  + Upload File

Terminal ▶ Run Copy this project

0 #!/usr/bin/python
1
2 import time
3 from Adafruit_I2C import Adafruit_I2C
4
5 # -----
6 # BMP085 Class
7 # -----
8
9 class BMP085 :
10     i2c = None
11
12     # Operating Modes
13     __BMP085_ULTRALOWPOWER = 0
14     __BMP085_STANDARD = 1
```

- ▶ Created to be used as teaching tools
- ▶ Development tools exist to ease development on Raspberry Pi as well

ARM



- ▶ Reduced Instruction Set Computing (RISC) Architecture

ARM



- ▶ Reduced Instruction Set Computing (RISC) Architecture
- ▶ Requires fewer transitions than CISC x86 Architectures

ARM



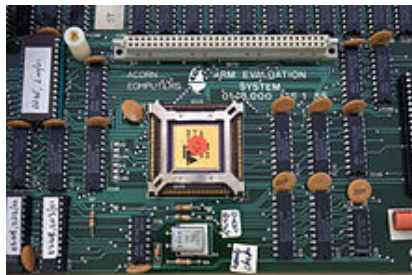
- ▶ Reduced Instruction Set Computing (RISC) Architecture
- ▶ Requires fewer transitions than CISC x86 Architectures
- ▶ Reduced cost, heat and power use

ARM



- ▶ Reduced Instruction Set Computing (RISC) Architecture
- ▶ Requires fewer transitions than CISC x86 Architectures
- ▶ Reduced cost, heat and power use
- ▶ Ideal for embedded systems

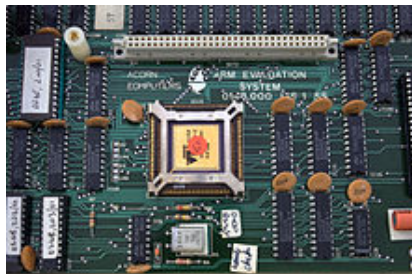
ARM



ARM Development Board

- ▶ Evaluate, benchmark and start software development

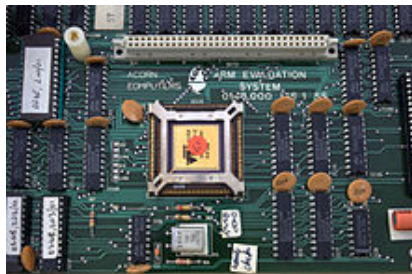
ARM



ARM Development Board

- ▶ Evaluate, benchmark and start software development
- ▶ Prototype and validate SoC IP

ARM



ARM Development Board

- ▶ Evaluate, benchmark and start software development
- ▶ Prototype and validate SoC IP
- ▶ Test custom logic

System on a Chip

- ▶ Integrated Circuit (IC) that integrates all components of a computer or other electronic system

System on a Chip

- ▶ Integrated Circuit (IC) that integrates all components of a computer or other electronic system
- ▶ Low power consumption makes them popular in embedded systems