19005 ARD

Introduction to chipKIT™ Development Platform

Who am I?

Keith Vogel
Senior Software Engineer
KeithV@Digilentinc.com

www.digilentinc.com

Please feel free to ask questions at any time.

Class Objectives

- Learn what the chipKIT™ Platform is
- How the chipKIT platform relates to Arduino
- Understand the chipKIT Core, the runtime system and supporting IDEs
- Know how to use MPIDE to create, compile and run sketches

Class Agenda

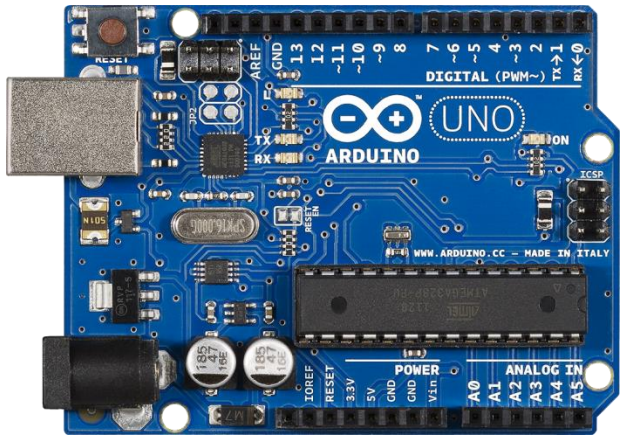
- What is Arduino?
- What is the chipKIT™ platform?
- Compatibility: chipKIT system vs Arduino system
- Overview of chipKIT hardware and software
- Using the MPIDE

WHAT IS ARDUINO?

What is Arduino?

"Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments."

from www.arduino.cc



Goals of Arduino System

- Ease of use
- Provide non-technical people (such as artists) the ability to use microcontrollers and embedded control in their projects
- Introduce beginners to embedded control using microcontrollers

What is Arduino?

- **An “open hardware”/“open source” embedded development platform**
  - Integrated Development Environment (IDE), compilers, linkers
  - Hardware platform specification
  - Hardware abstraction layer
  - Support libraries
- **Family of low cost microcontroller boards based**
- **An Italian company that manufactures and sells “Arduino” boards**

Hardware Abstraction Layer

- **Digital I/O uses logical pin numbers rather than ports and bits**
- **Analog input**
- **Pseudo-analog (PWM) output**
- **Timing functions**
- **External interrupts**
- **Communication Libraries**
  - UART, SPI, I²C™

Arduino Software

- **A simple IDE**
- **“Sketch” - the software program (artists, hobbyists, etc.)**
- **Preprocessing hides complexity**
- **Programming language: C++ compatible**
- **Software libraries**
  - Hardware abstracted peripheral support
  - Abstracted support classes

WHAT IS THE chipKIT™ PLATFORM?

What is the chipKIT™ Platform?

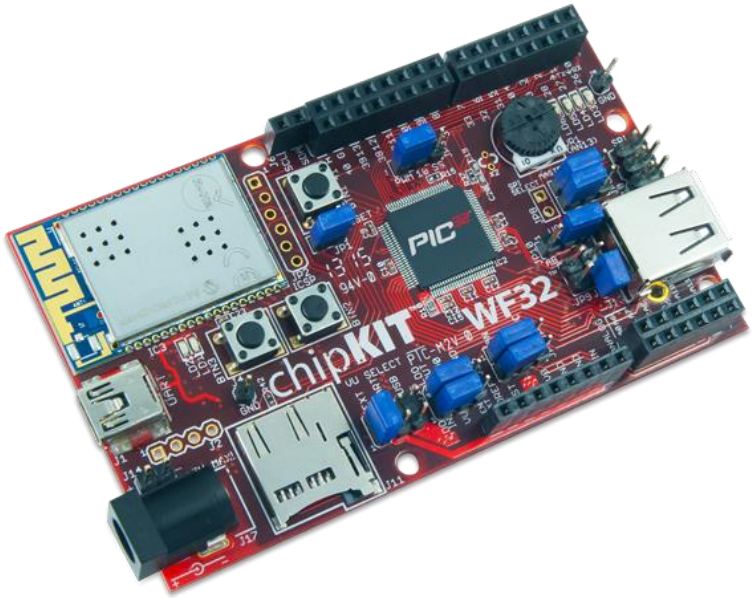
- Professional quality open hardware, open software, and support libraries using Microchip PIC32 MCUs
- An embedded chipKIT core supported by multiple IDEs (MPIDE, UECIDE, and MPLAB® X IDE)
- A full-featured tool suite (compilers, linkers, and debuggers)
- Arduino-compatible core functions and compatibility classes

Goals of chipKIT™ Platform

- Easy access to professional applications and libraries for new users
- Rapid prototype development tools
- Migration between easy to use IDEs and professional IDEs
- The same ease-of-use as Arduino when using Arduino-style IDEs
- Substantial source code compatibility with Arduino

Some chipKIT™ Boards

uC32



WF32

chipKIT™ Core - Software

- **The chipKIT Core**
  - Embedded core runtime functions
  - Arduino-compatible core runtime functions
  - Extensive peripheral libraries
  - Arduino-compatible libraries
  - Microchip chipKIT compiler and build tools
  - C/C++ Runtime library
- **Supported IDEs**
  - Ease of Use IDEs (no additional hardware needed)
    - **MPIDE**
    - **UECIDE**
    - **Arduino IDE (in the works)**
  - Professional IDE (requires ICSP™ hardware debugger)
    - **MPLAB® X IDE**

COMPATIBILITY: chipKIT™ SYSTEM VS. ARDUINO

Software Compatibility

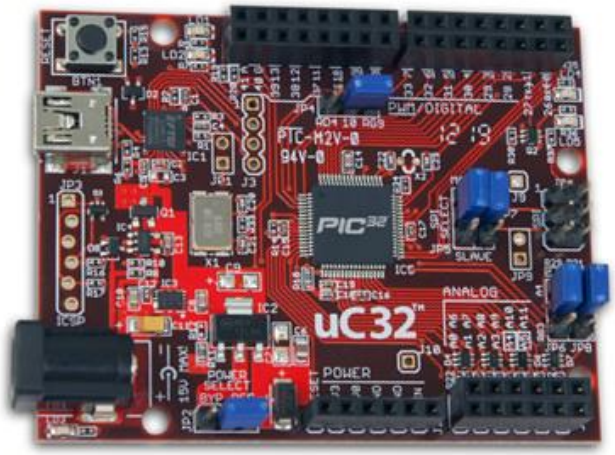
- Arduino version 023 core files and standard libraries ported to PIC32 platform
- Most sketches using only core files and standard libraries will work without change
- Arduino compatibility classes are layered on top of more advanced PIC32 libraries
- Timing-dependent code may have to be rewritten
- Libraries or sketches containing AVR-specific code must be rewritten to work with PIC32
- Work in progress to support compatibility with Arduino 1.6x standard

5V vs 3.3V Operation

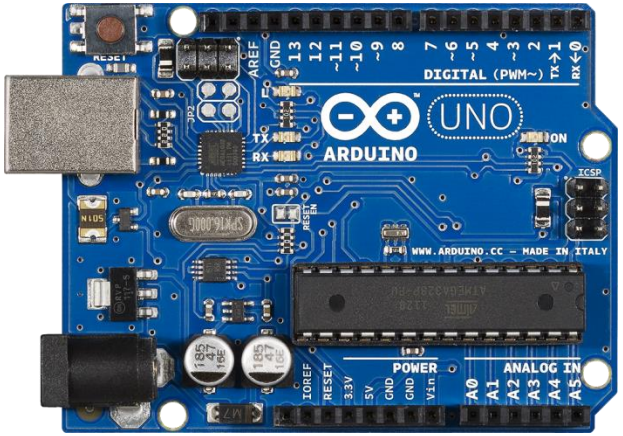
- **Most Arduino boards operate at 5V**
  - Exceptions include Arduino Due, Fio, Pro(168), LilyPad(168), LilyPad USB
- **All chipKIT™ boards operate at 3.3V**
  - Produce 3.3V outputs
  - PIC32MX boards: inputs 5V tolerant
  - PIC32MZ boards: inputs not 5V tolerant, accept 3.3V inputs only
- **Shields capable of accepting 3.3v inputs should work**
- **Shields that produce 5V outputs can't be used with PIC32MZ-based boards**

uC32 vs Uno

- PIC32MX340F512H
- 512K flash, 32K RAM, 1K EEPROM
- 42 I/O pins
- 2 UART
- 2 SPI
- 2 I²C™
- 12 10-bit A/D inputs
- 5 output compare/PWM



- ATmega328
- 32K flash, 2K RAM, 1K EEPROM
- 22 I/O pins
- 1 UART
- 1 SPI
- 1 I²C™
- 6 10-bit A/D inputs
- 6 output compare/PWM



IDE ENVIRONMENTS AND COMPATIBILITY

Development Environment

- MPIDE forked from the Java-based Arduino IDE, (open source)
- UECIDE forked from MPIDE, but substantially rewritten (open source)
- MPLAB® X IDE, NetBeans-based professional IDE (not open source)
- Cross platform (Windows, Mac, Linux)
- GCC based (Gnu C++ Compiler)

Microchip Tool Compatibility

- Start with easy-to-use IDEs for rapid prototype development or evaluation
- Migrate to MPLAB® X IDE at any time

OR

- Develop advanced code in MPLAB X IDE
- Deploy professionally developed code to the non-technical community with easy-to-use IDEs

MPLAB® Harmony vs. chipKIT™ Platforms

- **All chipKIT boards can be used directly with MPLAB X IDE and MPLAB Harmony**
- **Use Microchip MPLAB X IDE and XC32 C/C++**
- **Use with standard Microchip programmer/debugger tools**
  - PICkit™ 3 debugger, MPLAB ICD 3, MPLAB REAL ICE™ in-circuit emulator, etc.
- **MPLAB Harmony and chipKIT development environments are different and source incompatible**

chipKIT™ PLATFORM IN MORE DETAIL

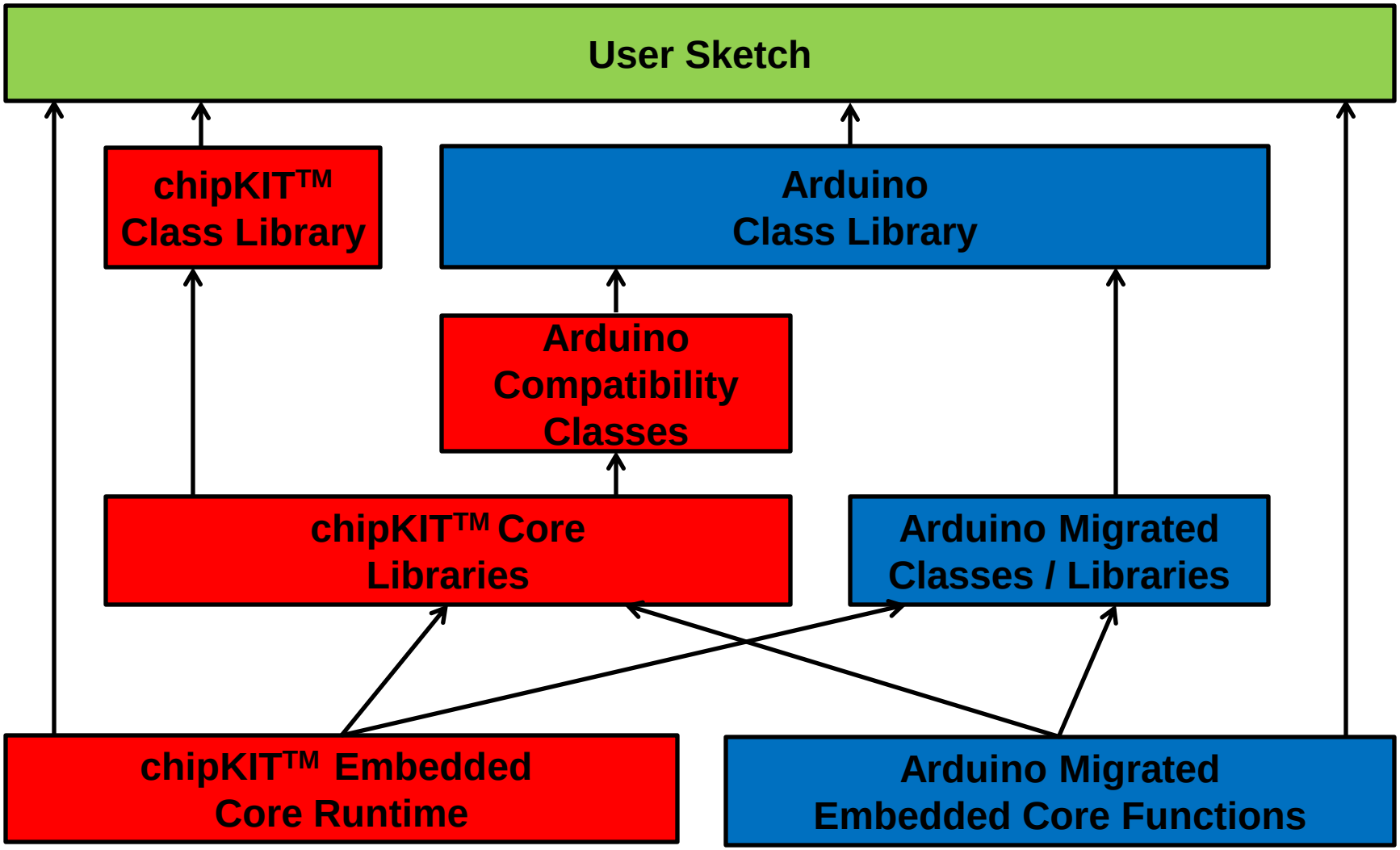
Key Platform Elements

- PIC32 Microcontroller
- chipKIT™ Embedded Core
- chipKIT Libraries
- Hardware Abstraction Layer
- Standard Runtime Libraries
- Bootloader
- Serial communications interface

PIC32 Microcontroller

- **32-bit MIPS processor core**
- **40MHz to 200MHz clock speeds**
- **Most instructions are single cycle**
- **Good set of standard peripherals**
  - UART, SPI, I²C™, Timers, etc.
- **Advanced peripherals**
  - USB OTG, 10/100MAC, CAN, etc.

chipKIT™ Core - Software



Arduino Compatible Libraries

- EEPROM
- Firmata
- Liquid Crystal (LCD character display)
- SPI
- Servo
- Stepper
- SD
- SoftSPI
- SoftwareSerial
- HardwareSerial
- OneWire
- Matrix
- Sprite

chipKIT™ Libraries

Core Libraries

- DSPI (SPI)
- DTWI (I²C™)
- SoftPWMServo (PWM)
- DEIPcK (Network)
- DEWFcK (WiFi)
- HTTPServer (HTTP)

Arduino Compatibility Classes

- Wire (I²C™)
- Ethernet (Network)

Bootloader

- **Simple IDEs**
  - Use bootloader to program the sketch onto the board
  - Communicate with bootloader via serial interface
  - No additional hardware required
- **MPLAB X® IDE**
  - Requires a hardware programmer/debugger
  - Programs the bootloader concurrently with the sketch
  - Also programs a debug executive for debugging

Serial Bootloader and Serial Monitor

- **Serial port is standard part of an Arduino/chipKIT™ board**
- **Used by bootloader to program sketch**
- **User can interact with the sketch at runtime**
  - MPIDE provides a serial monitor window where user can view output and provide input

Bootloader Programming

MPIDE passes Avrdude the sketch HEX file

MPIDE spawns Avrdude

HEX

Avrdude

Virtual COM Port 115200 baud

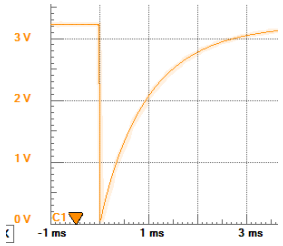
STK500v2 Protocol

FTDI Serial to USB Driver

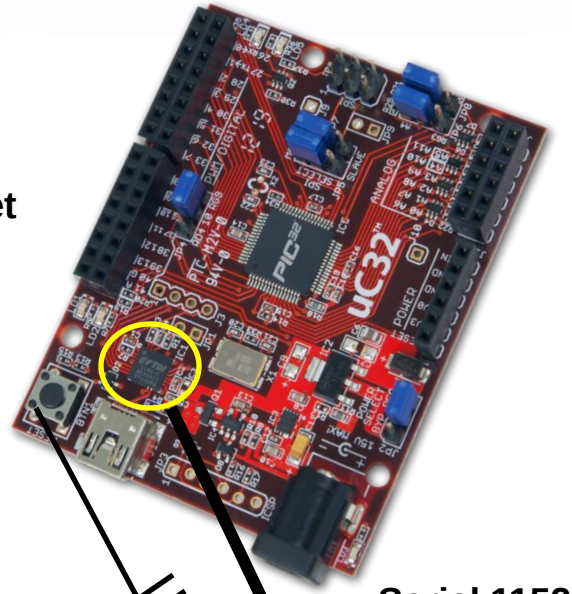
USB

Artifact of USB Enumeration causes the FTDI chip to pulse DTR

Bootloader programs the sketch on reset

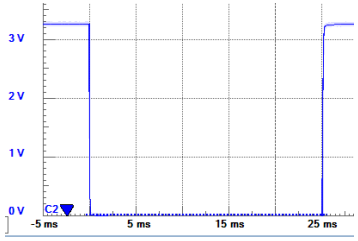


Reset / MCLR

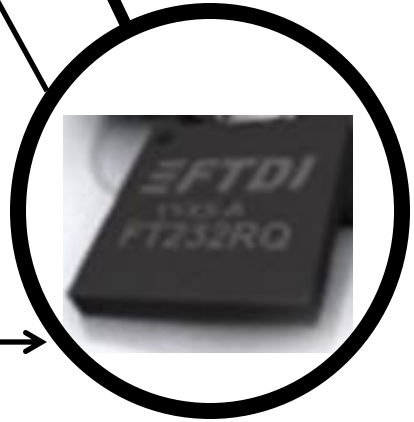


.1uF

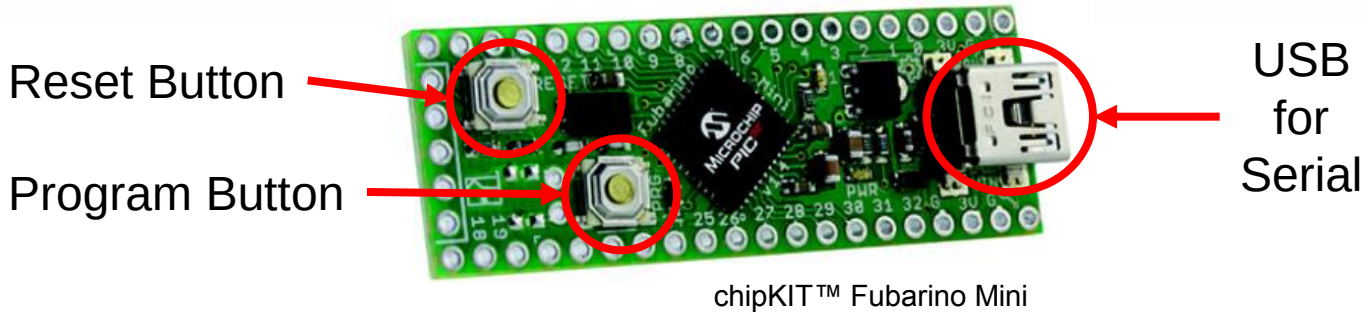
Serial 115200 Baud



DTR



USB for Serial



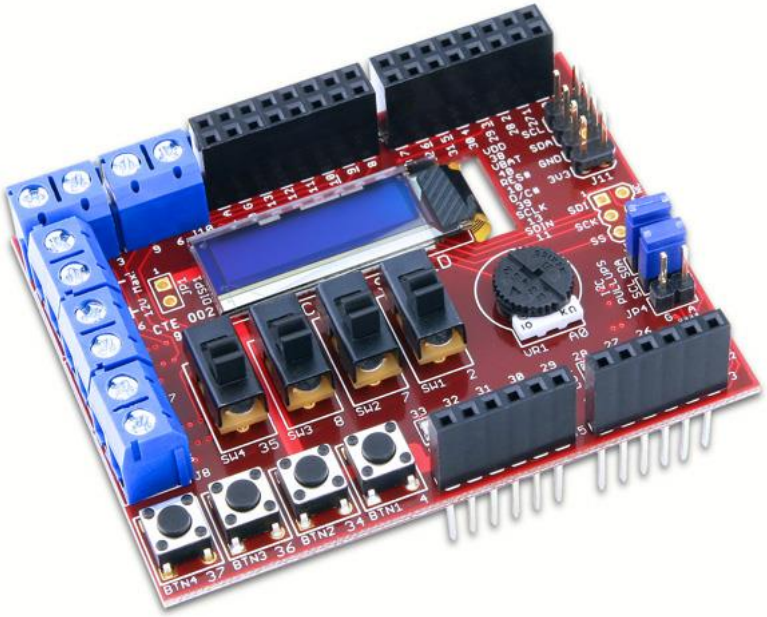
chipKIT™ Fubarino Mini

- **Some boards do not use an FTDI chip and use the PIC32 USB controller directly**
- **Reduces cost and size of board**
- **Requires install of custom USB serial device .inf**
  - Links custom serial device class to OS provided USB serial driver
- **No DTR, no auto reset; USB for Serial requires a program button**
- **Hold program button and then press reset, puts bootloader in program mode**
- **Also used to support the Serial Monitor**

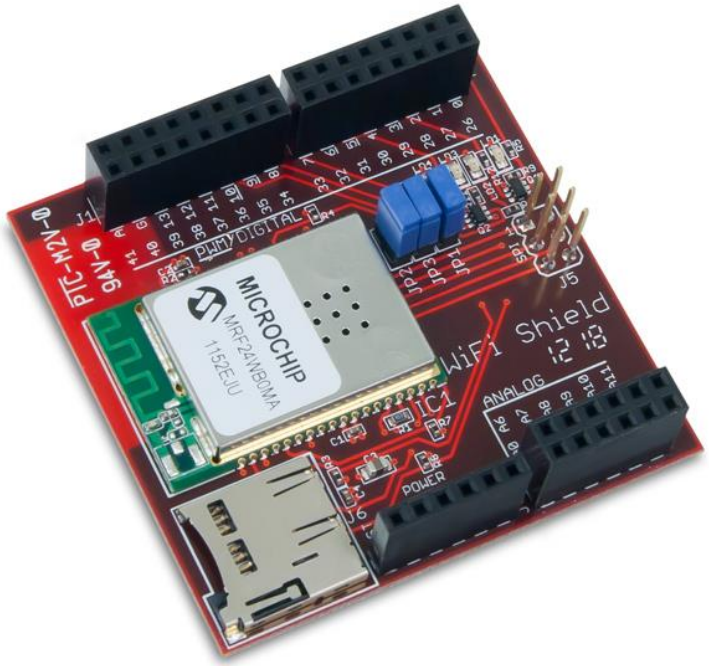
Shield Interface

- **100 mil pin header connectors**
- **Boards stack vertically**
  - Base board has female headers
  - Shield board has male headers
- **Peripheral functions assigned to specific pins**
- **chipKIT™ shields have an extra row of headers to provide more I/O**

chipKIT™ Shields

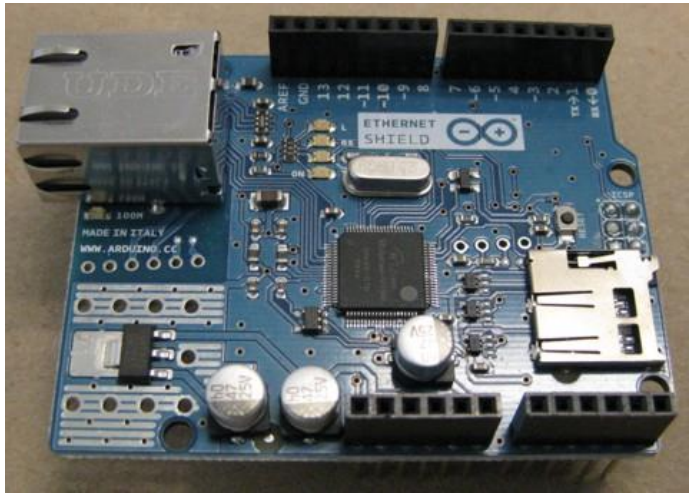


Basic I/O Shield

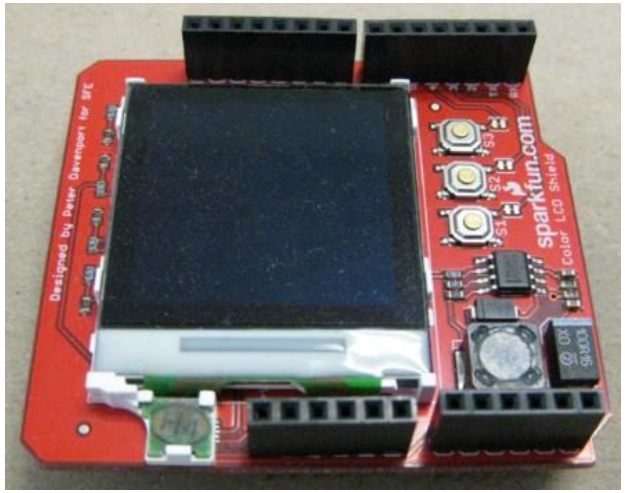


Wi-Fi® Shield

Some Arduino Shields



Ethernet Shield



Graphic
Display Shield

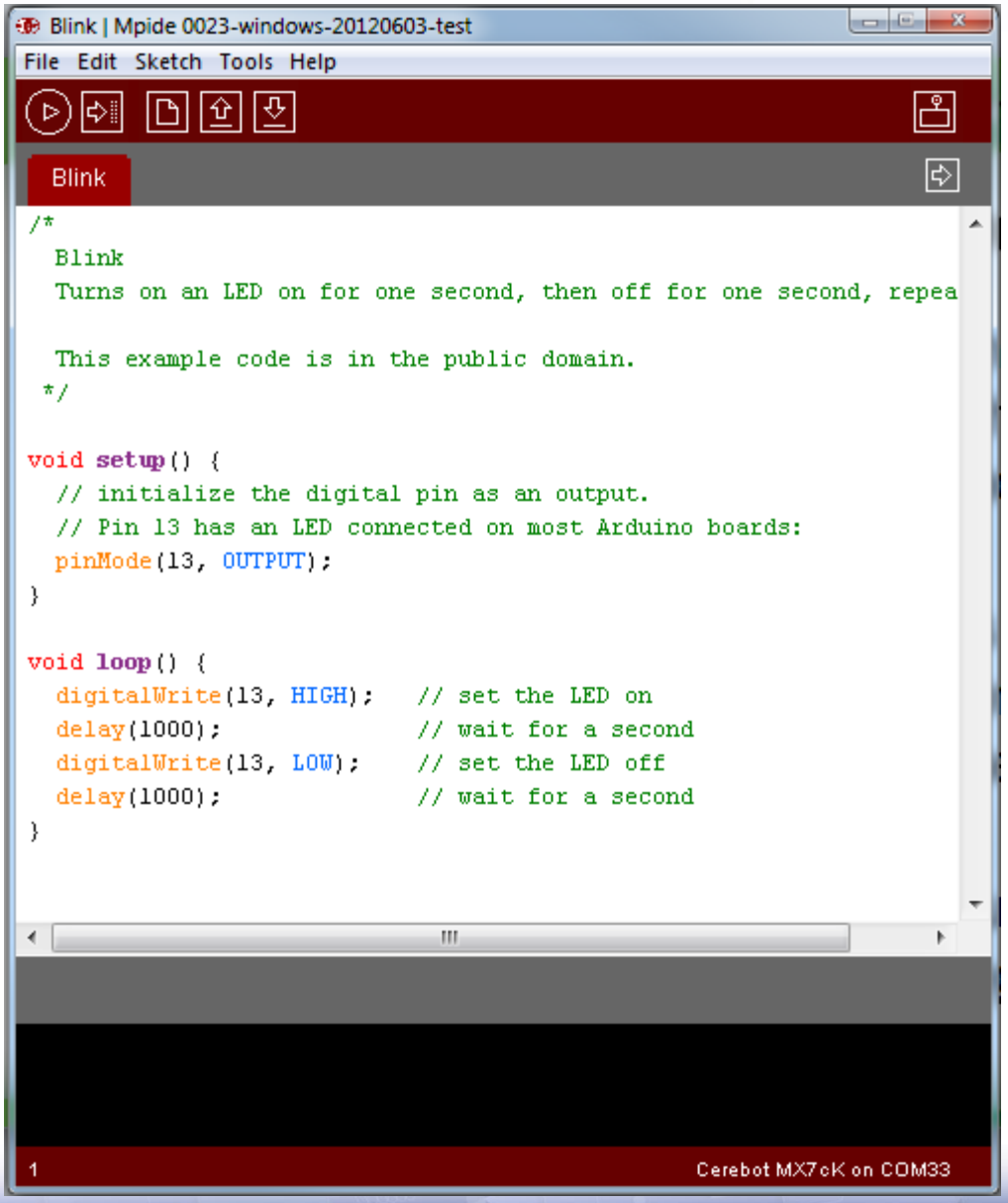
A large number of Arduino shields are suitable for use with chipKIT™ boards

Motor Drive Shield

PREPARATION FOR LAB: USING THE MPIDE

Lab Objectives

- Walkthrough of the IDE
- Programming model
- Use of core functions
- Use of standard libraries
- Building and running sketches



Programming Model

- **A program is called a sketch**
- **Language used is C++**
- **Embedded program event loop is built into the system**
- **Two required functions**
  - `setup()` – one time initialization
  - `loop()` – called repeatedly by system

Programming Model

User program (sketch) implements two required functions.

```
void setup() {
     // called ONCE at the beginning
}

void loop() {
     // called repeatedly
}
```

Programming Model

Behind the Scenes (Preprocessing)

```
int main(void) {
      init();   //system initialization

      setup();  //user initialization

      while (1) {
            loop();   //sketch behavior
            }
      return 0;
}
```

Core Runtime Functions

- `pinMode(pin, dir)`
  - Sets pin direction and drive type
- `digitalRead(pin)`
  - Reads the state of a digital pin
- `digitalWrite(pin, val)`
  - Sets a digital pin to specified state
- `delay(ms)`
  - Delay the specified number of milliseconds

Example – Blink an LED

```
int ledPin = PIN_LED1;          //led on pin 13

void setup() {
    pinMode(ledPin, OUTPUT);  //make pin an output
}

void loop() {
    digitalWrite(ledPin, HIGH);  //led on
    delay(1000);                 //wait one second
    digitalWrite(ledPin, LOW);   //led off
    delay(1000);                 //wait one second
}
```

Core Runtime Functions

- `analogRead(pin)`
  - Returns A/D converter value for specified pin
- `analogWrite(pin, val)`
  - Pseudo-analog output using PWM. Sets output duty cycle to specified value

Example – Dim an LED

```
int pinLed = 9;       //assume LED on pin 9
int pinPot = A0;      //assume pot on analog 0

   void setup() {     //nothing needed
   }

   void loop() {
      int val;
        val = analogRead(pinPot);
        analogWrite(pinLed, val/(1024/256));

        delay(1000);
   }
```

Hardware Serial

- `Serial.begin(baud)`
  - Initialize the UART and set the baud rate
- `Serial.print(val)`
  - Print the specified value to UART
- `Serial.read()`
  - Read characters from UART

Example

```
void setup() {
    Serial.begin(9600);    //init UART
}

void loop() {
    Serial.println();
    Serial.println("Hello World!");
    for (i = 1; i <= 10; i++) {
        Serial.print(" i = ");
        Serial.println(i, DEC);
    }
    delay(5000);       //wait five seconds
}
```

Lab Activity

- **Open the lab handout and do the lab exercises**

Lab Summary

- Features/use of IDE
- Structure of a Sketch
- Available runtime facilities
- Use of libraries
- Getting output from a sketch

Class Summary

- What is the chipKIT™ platform?
- How does it compare to Arduino?
- chipKIT system hardware features
- MPIDE and runtime software features
- Write, build, download and execute sketches

Resources

- **Help and examples in MPIDE**
- **chipKIT™ Platform web site**
  http://chipkit.net
- **chipKIT support forum**
  http://chipkit.net/forum
- **Digilent sites**
  http://digilentinc.com/chipkit
  http://learn.digilentinc.com
- **Developer Help**
  http://microchip.wikidot.com

chipKIT™ Cores and IDEs

- **chipKIT Core**

  https://github.com/chipKIT32/chipkit-core

- **MPIDE**

  https://github.com/chipKIT32/chipKIT32-MAX
  http://chipkit.s3.amazonaws.com/index.html

- **UECIDE**

  http://uecide.org

- **MPLAB X® IDE**

  http://www.microchip.com/mplab

LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by "open source" software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided "AS IS." Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE. PLEASE BE AWARE THAT IMPLEMENTATION OF INTELLECTUAL PROPERTY PRESENTED HERE MAY REQUIRE A LICENSE FROM THIRD PARTIES.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

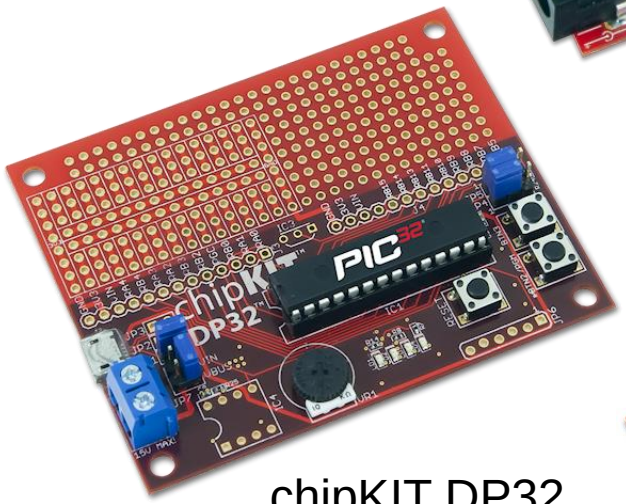
GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

APPENDIX

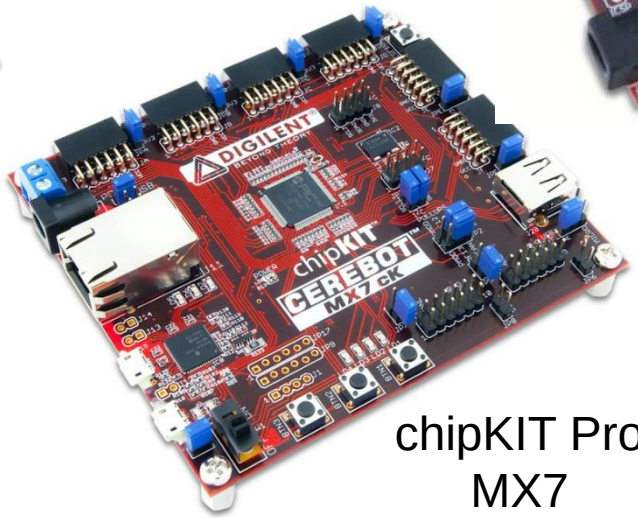
More chipKIT™ Boards

chipKIT Wi•FIRE



chipKIT DP32

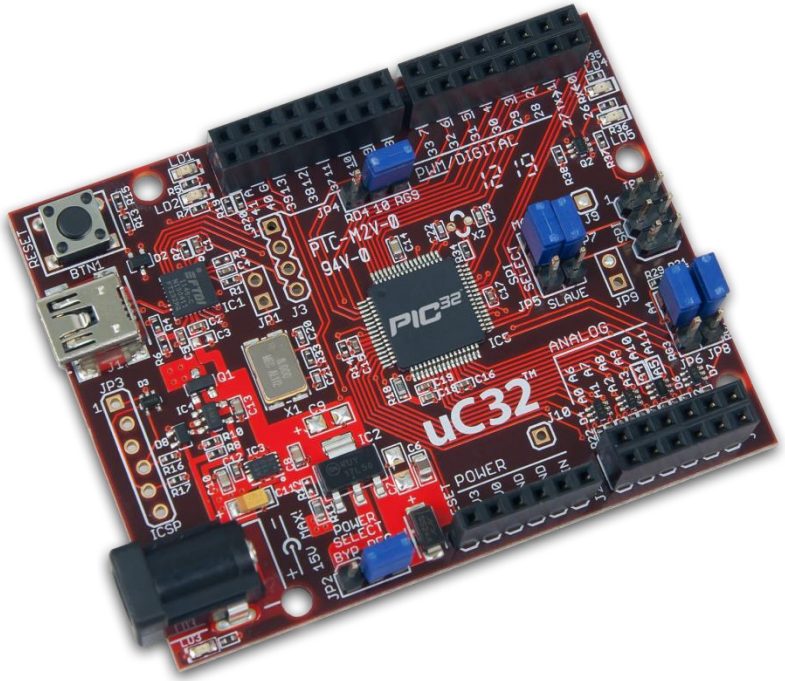
chipKIT MAX32



chipKIT Pro MX7

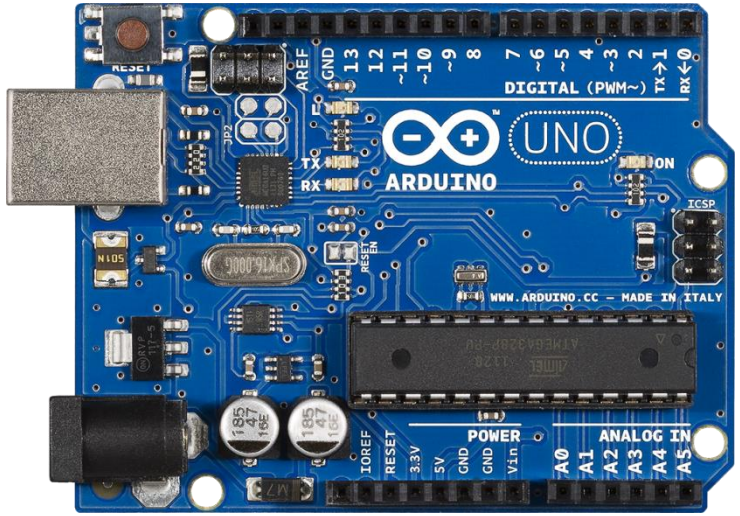
chipKIT™ uC32 Board

- **Microchip PIC32MX340F512H**
  - 80 Mhz 32-bit MIPS
  - 512K Flash
  - 32K SRAM
- **Arduino Uno form factor**
- **42 available I/O pins**
- **Two user LEDs**
- **12 analog inputs**
- **75mA typical operating current**
- **20V input voltage (maximum)**
- **0V to 3.3V analog input voltage range**
- **+/-18mA DC current per pin**



Arduino Uno

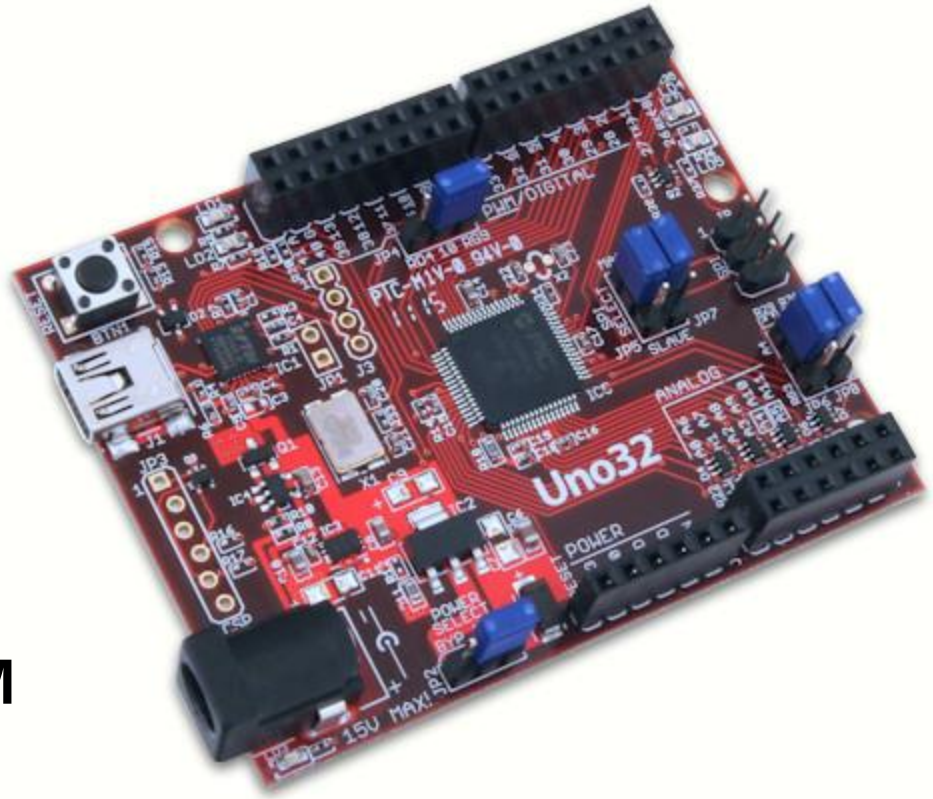
- ATmega328
- 32K flash, 2K RAM, 1K EEPROM
- 22 I/O pins
- 1 UART
- 1 SPI
- 1 I^2C™
- 6 10-bit A/D inputs
- 6 output compare/PWM



chipKIT™ Uno32

(replaced by a uC32)

- PIC32MX320F128H
- 128K flash, 16K RAM
- 42 I/O pins
- 2 UART
- 1 SPI
- 2 I^2C™
- 12 10-bit A/D input
- 5 output compare/PWM

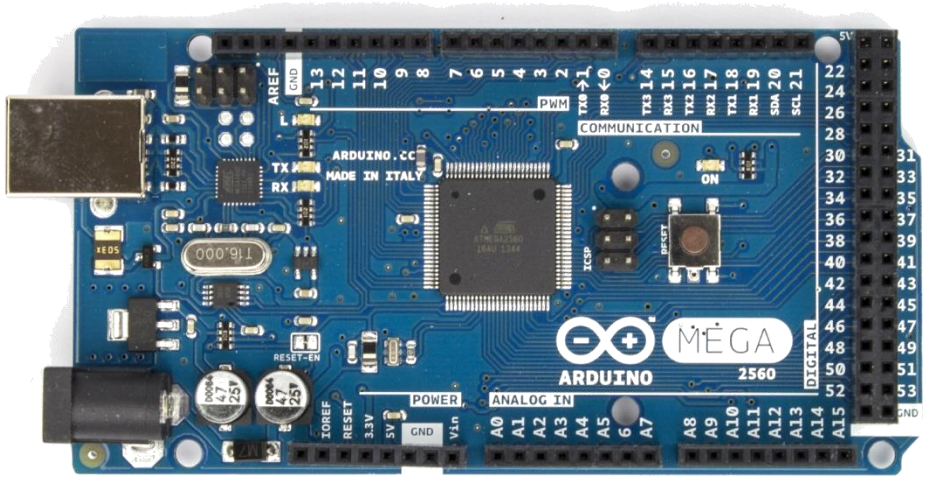


chipKIT™ Max32

- PIC32MX795F512L
- 512K flash, 128K RAM
- 83 I/O pins
- 4 UART, 1 SPI, 2 I²C™
- 16 10-bit A/D inputs
- 5 output compare/PWM
- USB 2.0 OTG controller
- 10/100 Ethernet MAC
- Dual CAN controllers

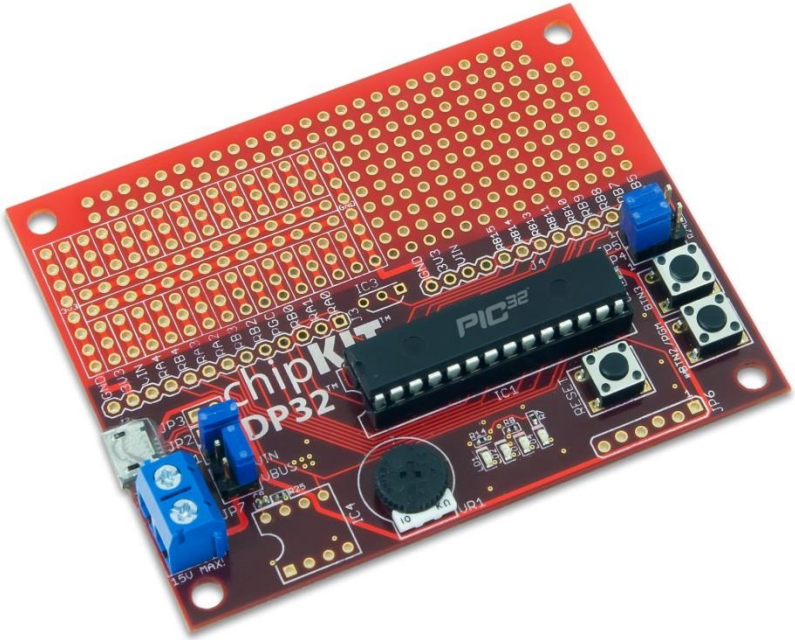
Arduino Mega 2560

- ATmega2560
- 256K flash, 8K RAM, 4K EEPROM
- 70 I/O pins
- 4 UART
- 1 SPI
- 1 I^2C™
- 16 10-bit A/D input
- 16 output compare/PWM
- 16 pins left unconnected



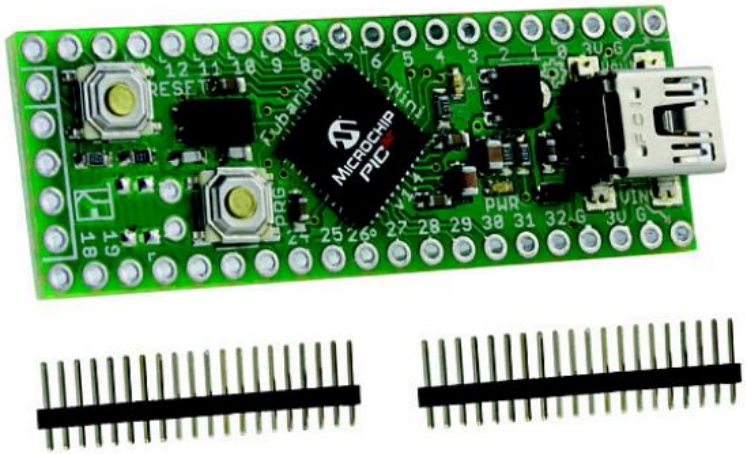
chipKIT™ DP32 Board

- Microchip PIC32MX250F128B
  - 40/50 MHz 32-bit MIPS
  - 128K Flash
  - 32K SRAM
- 19 available I/O pins
- Four LEDs, two push-buttons
- 9 analog inputs
- Potentiometer
- Wire-wrap prototype area
- Provision for an SPI EEPROM and an analog temperature sensor
- Mounting Hole compatible with Hammond 1591XXSSBK project box



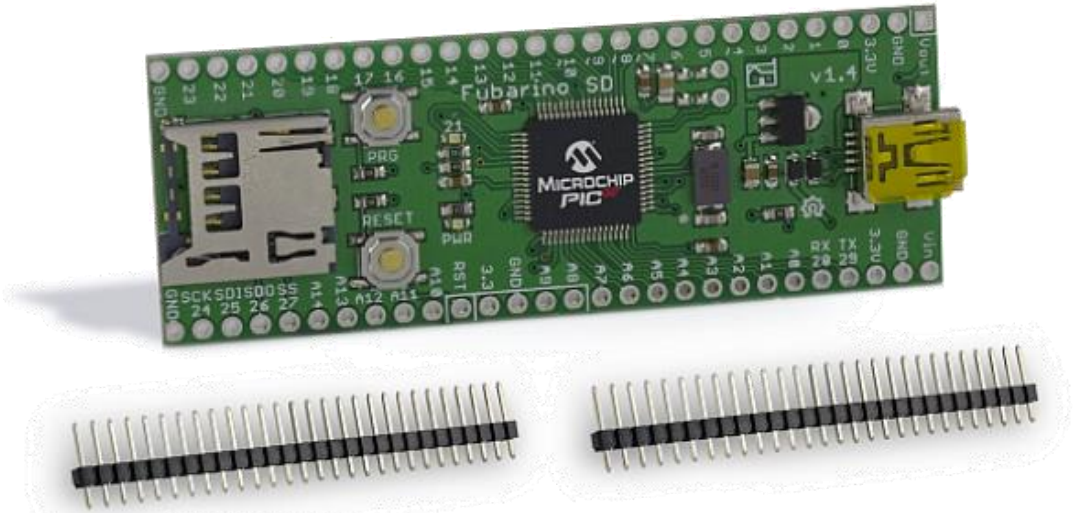
chipKIT™ Fubarino Mini

- PIC32MX250F128D-50I/ML
  - 48MHz operation
  - 128K flash
  - 32K RAM
- DIP form factor
- Max 33 I/O pins (normally 27)
- Pads for 32 KHz crystal
- Comes with headers (not installed) for easy mounting on a breadboard
- Two buttons: RESET for resetting the board, and PRG for getting into bootloader mode and user application use



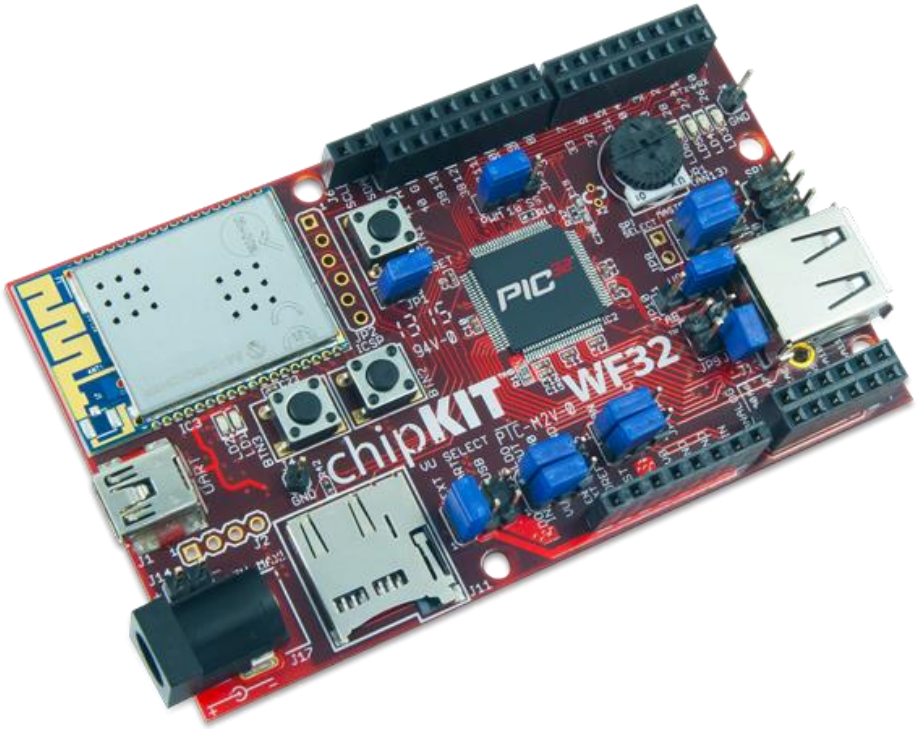
chipKIT™ Fubarino SD

- PIC32MX795F512H
  - 80MHz operation
  - 512K Flash
  - 128K RAM
- DIP form factor
- uSD card connector
- 45 available I/Os
- 15 analog inputs
- Two UART
- Comes with headers (not installed) for easy mounting on a breadboard
- Two buttons: RESET for resetting the board, and PRG for getting into bootloader mode and user application use



chipKIT™ WF32 Board

- Microchip PIC32MX695F512L
  - 80 Mhz 32-bit MIPS
  - 512K Flash
  - 128K SRAM
- 802.11g Wi-Fi®, MRF24WG0MA
- USB Host/Device
- uSD card connector
- Uno32 shield compatible
- 42 available I/O pins
- Four LEDs, 2 Push buttons
- Potentiometer
- 12 analog inputs
- Switching power supply

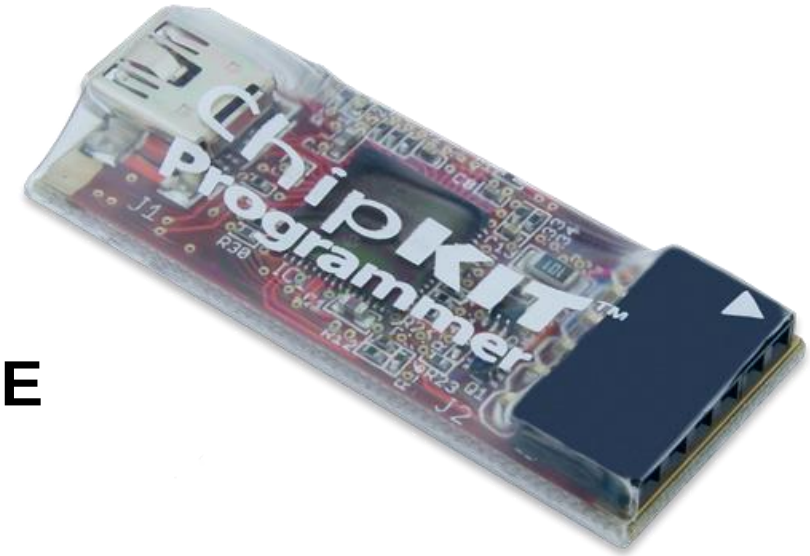


chipKIT™ Wi•FIRE™ Board

- **PIC32MZ2048EFG**
  - 200MHz MIPS microAptive core
  - 2048K Flash
  - 512K RAM
- **802.11/g Wi-Fi®, MRF24WG0MA**
- **USB 2.0 High Speed Host/Device**
- **uSD card socket**
- **Uno32 shield compatible with Arduino R3 enhancement**
- **42 available I/O pins**
- **Four LEDs, two push-buttons**
- **Potentiometer**
- **Switching power supply**

chipKIT™ PGM Programmer

- Provides in-system programming and debugging for Microchip PIC® MCU-based microcontroller boards
- Intended for use with chipKIT boards
- Works with the MPLAB® IDE and MPLAB X IDE

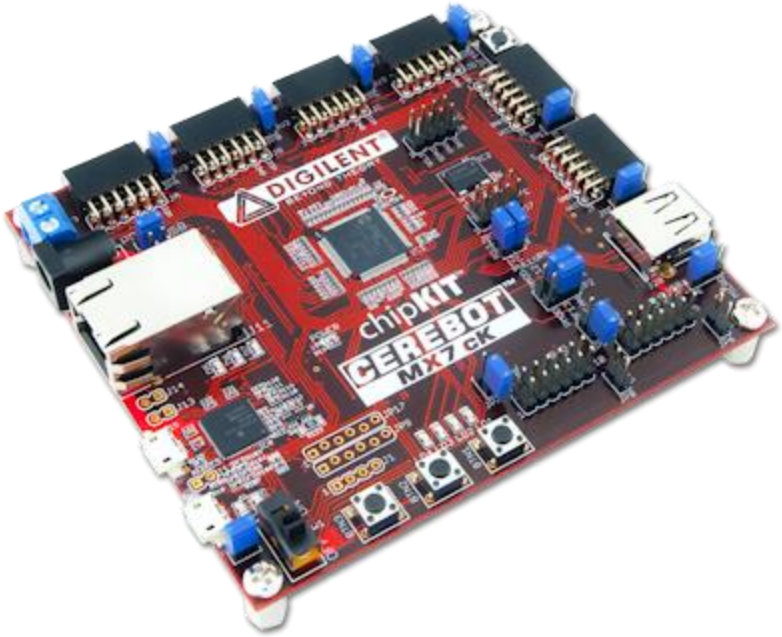


chipKIT™ Pro MX4 Board

- **PIC32MX460F512L**
  - 80 MHz operation
  - 512K flash memory
  - 32K RAM memory
- **74 available I/O pins, 9 Pmod, 2 I²C™, 8 servo connectors, 2 push buttons, 4 LEDs**
- **2 UART, 2 SPI, 2 I²C**
- **5 output compare, PWM**
- **14 10-bit A/D inputs**
- **MPLAB® IDE compatible licensed debugger**

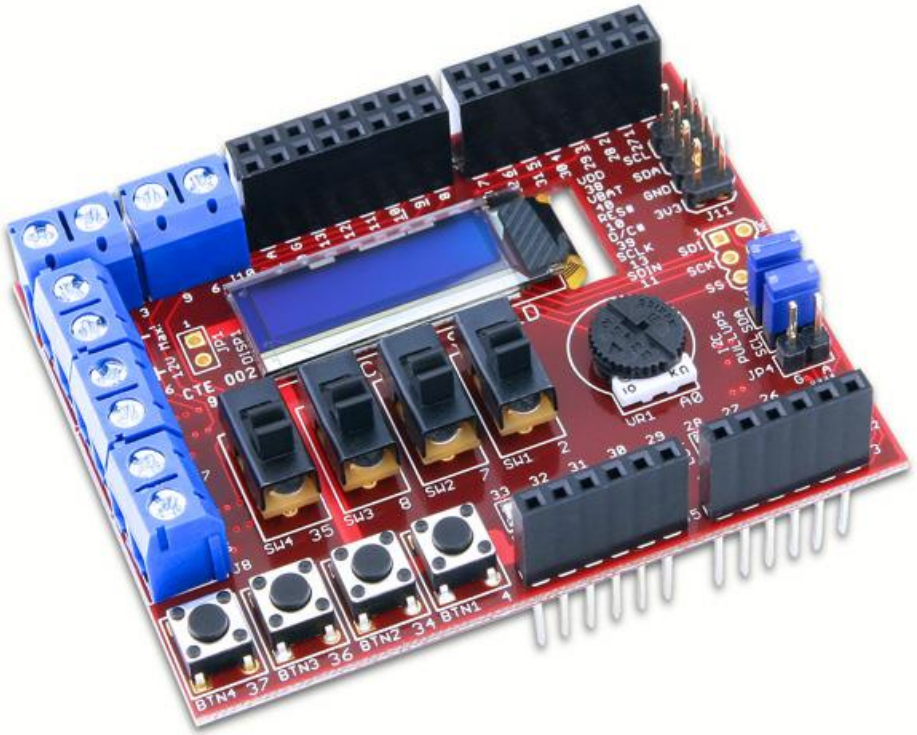
chipKIT™ Pro MX7 Board

- **PIC32MX795F512L**
  - 80 MHz operation
  - 512K flash
  - 128K RAM
- **64 available I/O pins**
- **6 Pmod connectors**
- **10/100 Ethernet, 2 CAN, USB 2.0 OTG Host/Device**
- **2 UART, 3 SPI, 2 I²C™**
- **5 output compare/PWM**
- **12 10-bit analog input**
- **MPLAB® IDE compatible licensed debugger**



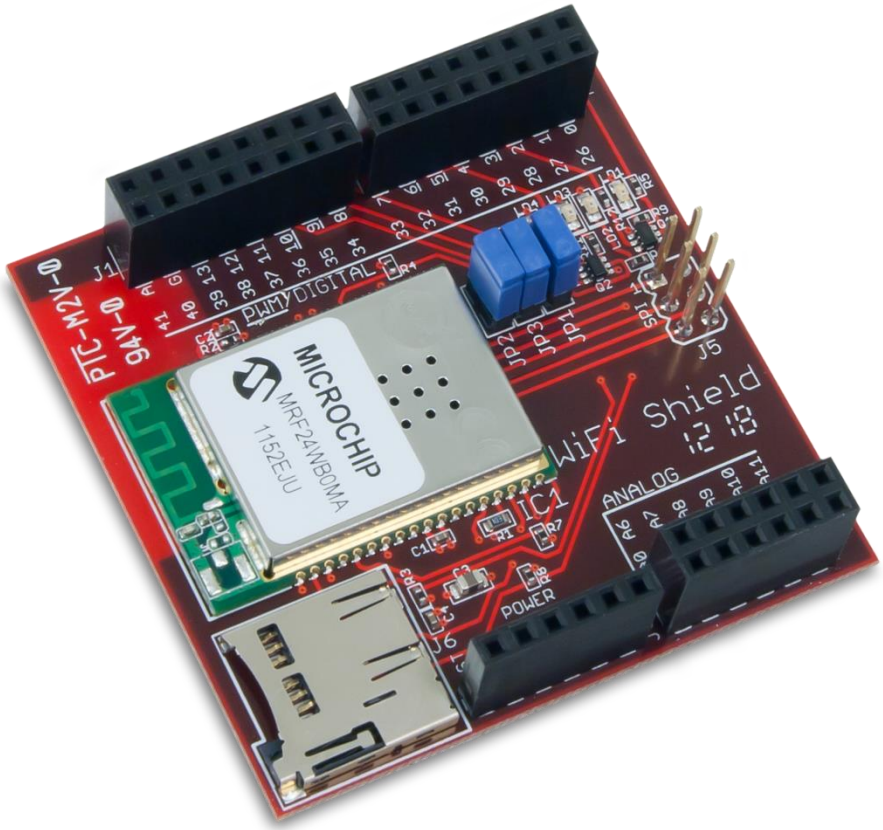
Basic I/O Shield

- Uno32 form factor
- 128x32 OLED display
- Four buttons
- Four slide switches
- Eight LEDs
- Four open drain FETs
- I²C™ EEPROM
- I²C Temp sensor
- Potentiometer



chipKIT™ WiFi Shield Board

- IEEE 802.11b-compliant RF transceiver
- serialized unique MAC address
- 1 and 2Mbps data rates
- IEEE 802.11b/g/n-compatible
- integrated PCB antenna
- range: up to 400m (1300 ft.)
- radio regulation certification for the FCC, IC, ETSI, and ARIB
- Wi-Fi® certified (WFA ID: WFA7150)
- micro SD card connector
- four LEDs



Network Shield

- Expands I/O on Max32
- USB Host & Device connectors
- 10/100 Ethernet PHY and connector
- Dual CAN transceivers and connectors
- Dual I²C™ connectors
- I²C EEPROM
- 32Khz oscillator

The Network Shield in combination with the Max32 provides access to all of the features of the PIC32MX795F512L.

chipKIT™ Motor Shield Board

- Two h-bridge DC motor channels
- Quadrature encoder feedback inputs
- Four RC servo outputs
- Four low side N-FET outputs
- I²C™ I/O expander with four LEDs, two push buttons, two jumpers
- Fits Uno32 style shield connector footprint