

Microprocessors & Microcontrollers

ME490A/B

Dr. C. Alex Simpkins

SDSU

Dept. of Mechanical Engineering

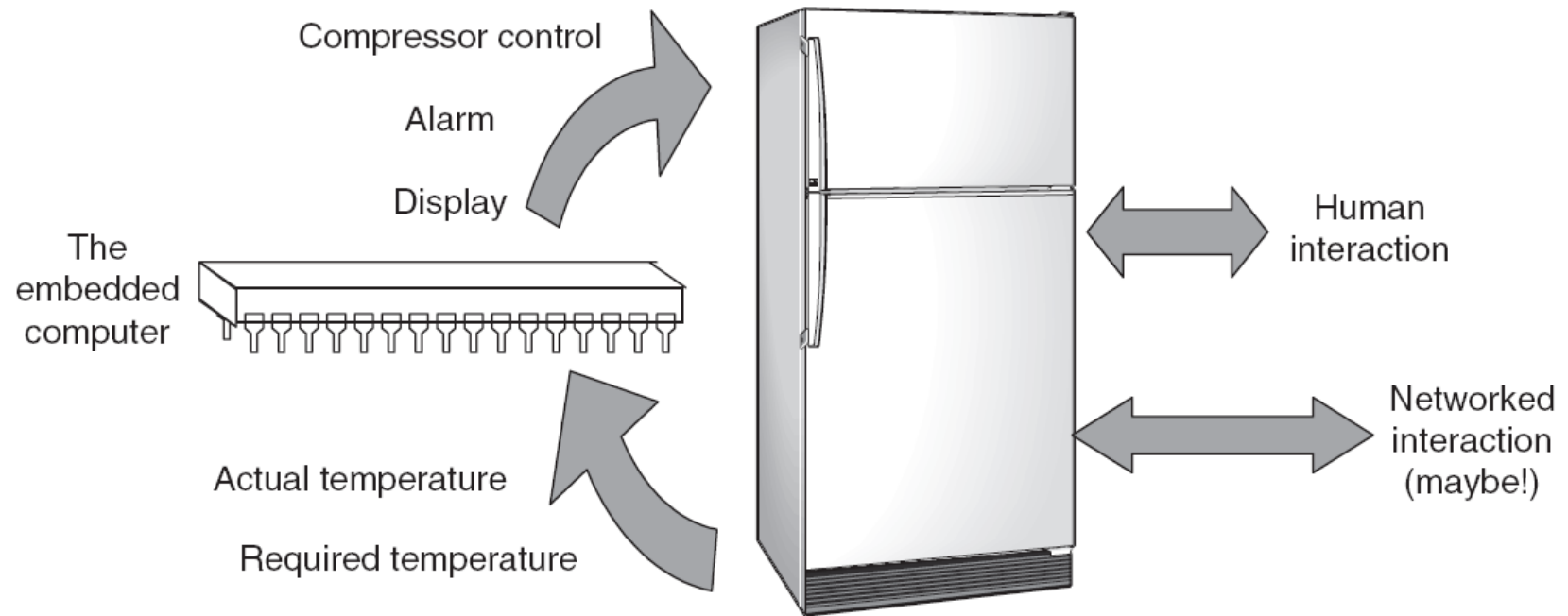
What is a Microcontroller?

- A microcontroller is a kind of miniature computer found in all kinds of gizmos



- Generally speaking, if a device has buttons and a digital display, chances are it also has a programmable microcontroller brain.

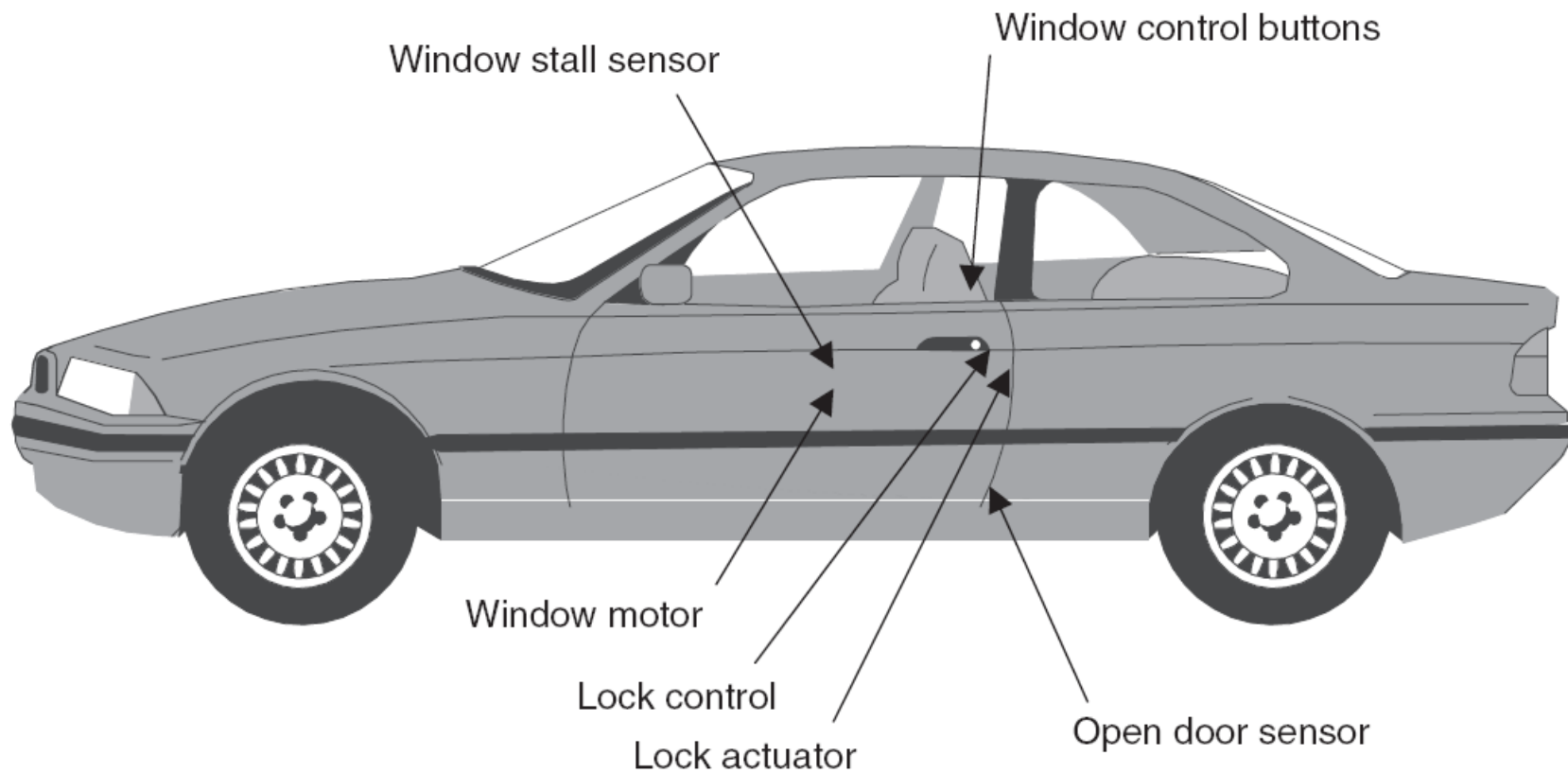
Examples: Refrigerator



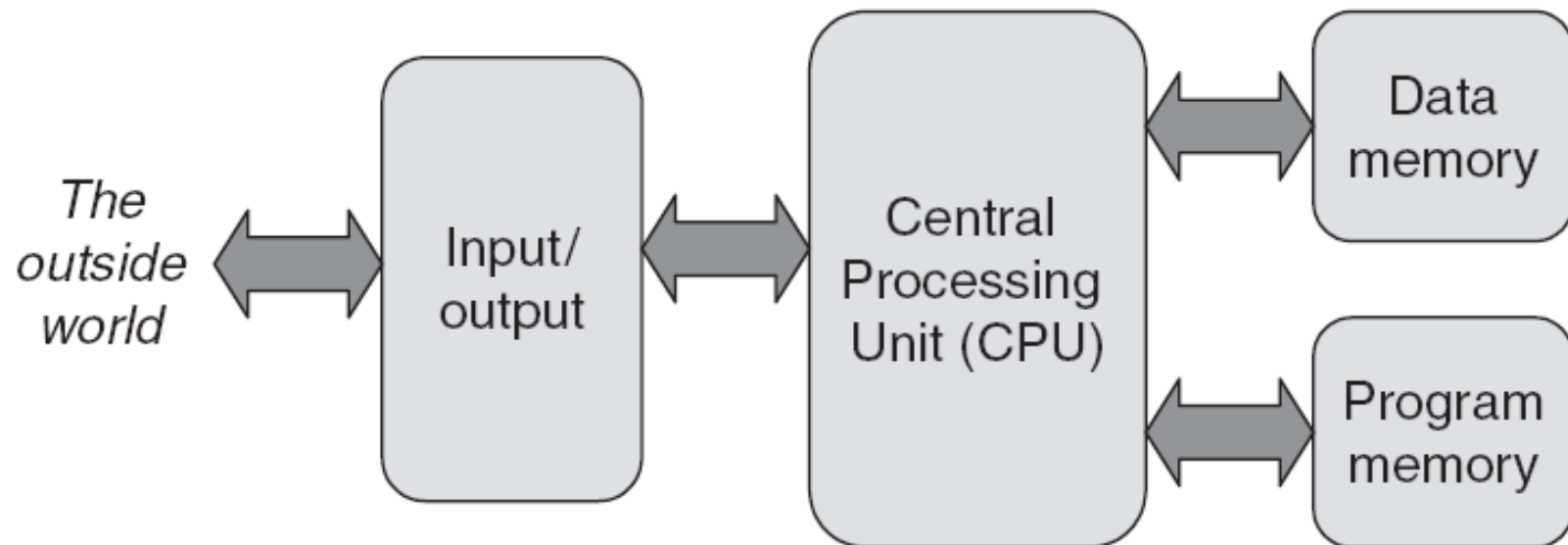
Microprocessor vs. Microcontroller

- A **microprocessor** is the “brain” of a computer system
- Generally referred to as the central processing unit (CPU), the microprocessor by itself is practically useless
- To be useful, one must have means of communicating with it using input and output devices
- One must also add memory (ROM and RAM) so that the system can be programmed.

Examples: Car Door



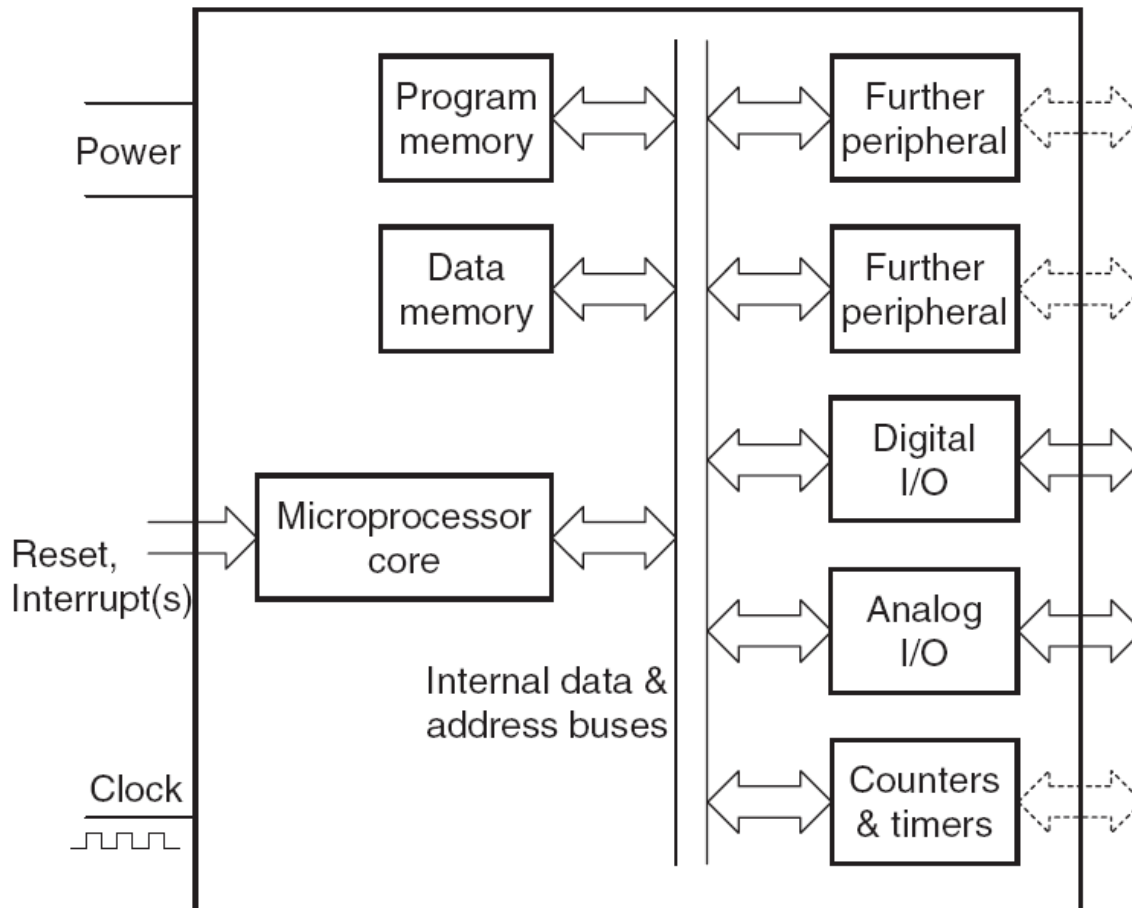
Computer Essentials



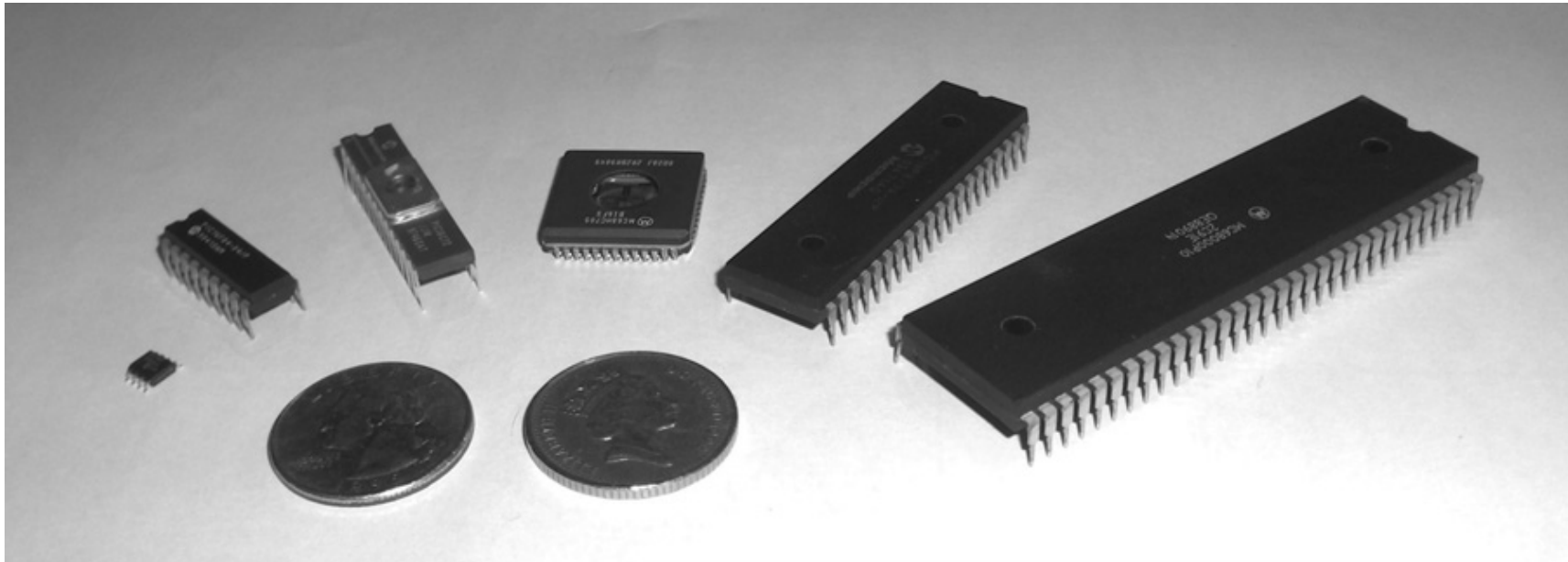
Microprocessors and Microcontrollers

- A **microprocessor** is a processor on one silicon chip.
- **Microcontrollers** are used in embedded computing.
- A **microcontroller** is a **microprocessor** with added circuitry.

Microcontrollers

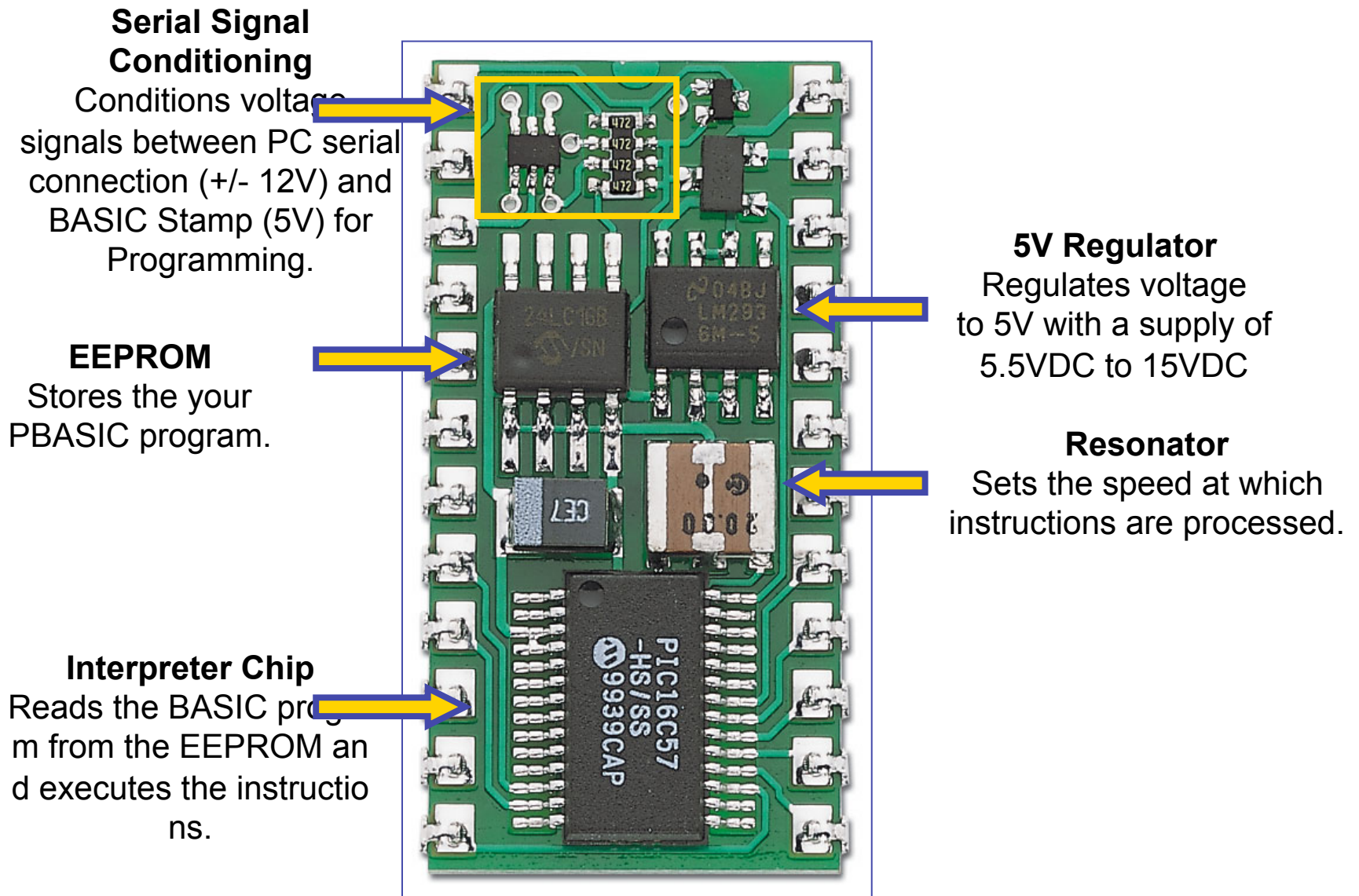


Microcontroller Packaging and Appearance



From left to right: PIC 12F508, PIC 16F84A, PIC 16C72, Motorola 68HC05B16, PIC 16F877, Motorola 68000

BASIC Stamp Module Components



Serial Signal Conditioning

Conditions voltage signals between PC serial connection (+/- 12V) and BASIC Stamp (5V) for Programming.

EEPROM

Stores the your PBASIC program.

Interpreter Chip

Reads the BASIC program from the EEPROM and executes the instructions.

5V Regulator

Regulates voltage to 5V with a supply of 5.5VDC to 15VDC

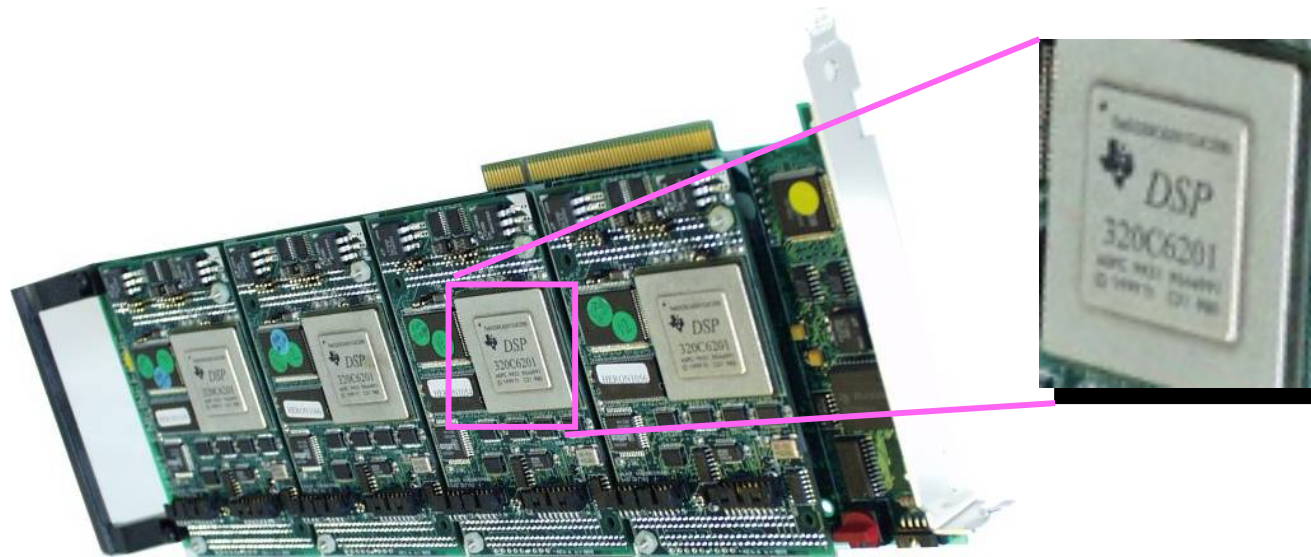
Resonator

Sets the speed at which instructions are processed.

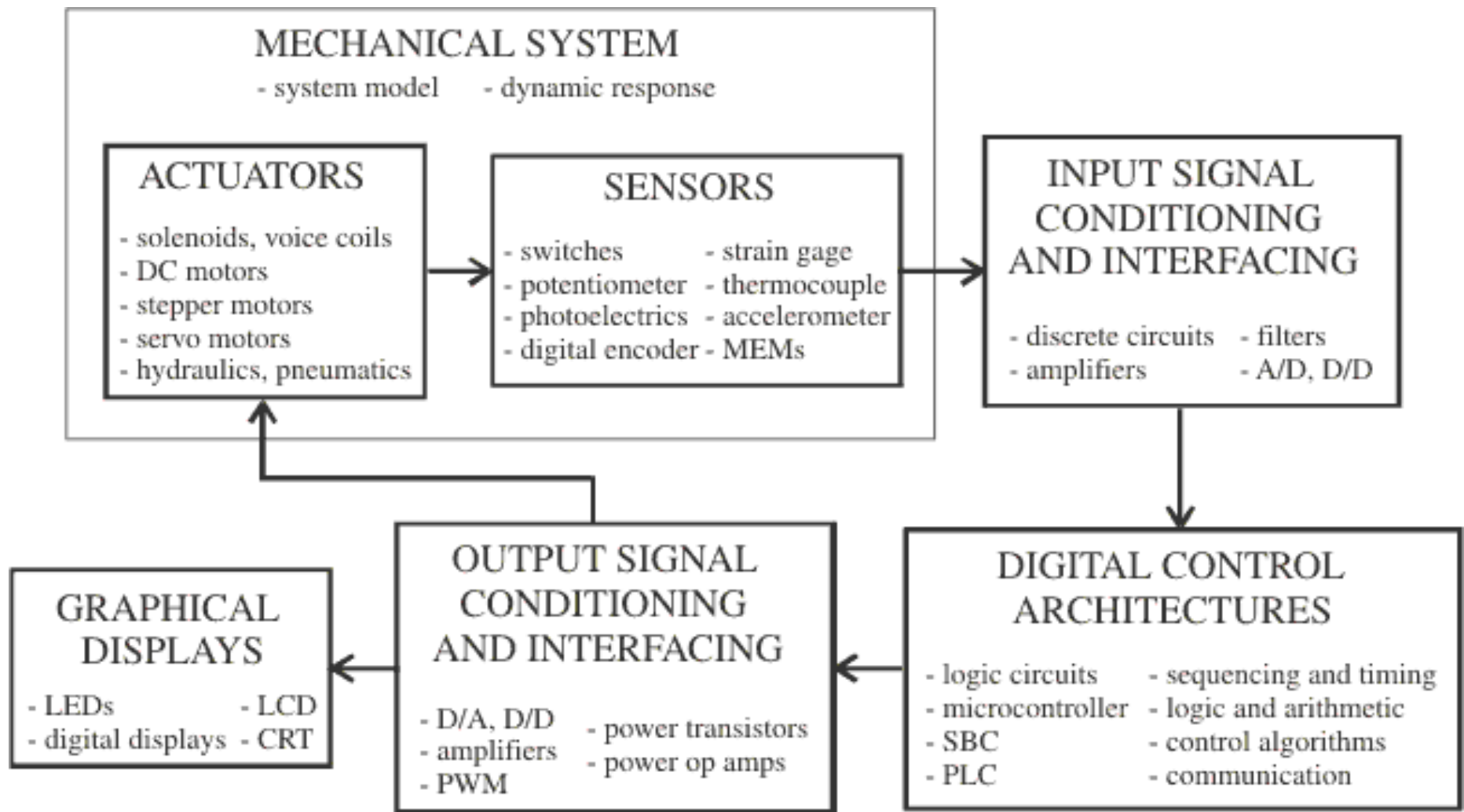
Digital Signal Processors

Digital Signal Processors (DSP's) are types of microcontrollers that are more specialized for a particular application. DSP represent around 20% of the total MCU market.

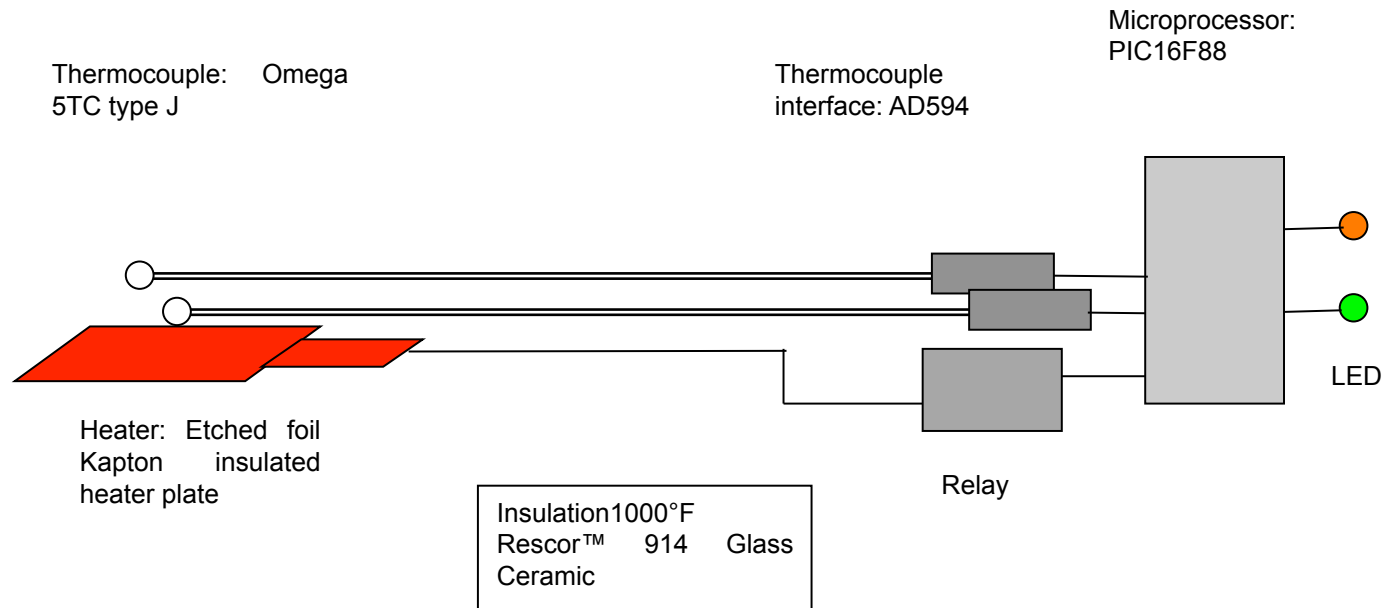
DSP's are particularly well suited to real-time operations in which a data stream, video, audio, etc. is modified in some way as it is passed along to another device or component



Mechatronic system components

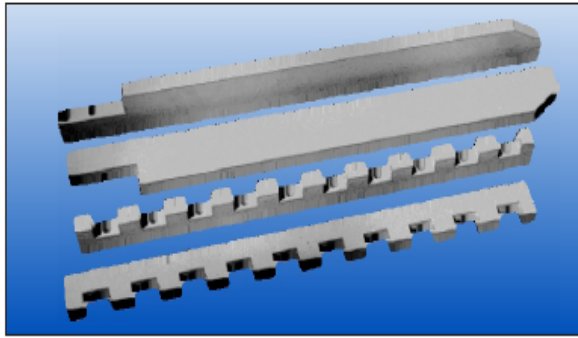


Overall schematic diagram of medical mini-heater

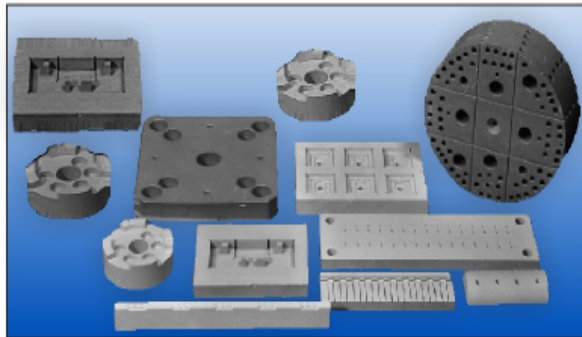


Insulation 1000°F

Rescor™ 914 Glass Ceramic



Use Temperature °F (Max.)	1000	Density (gm/cc)	2.6
Compressive Strength (psi)	40,000	Dielectric Strength (volt/ mil)	480
Flexural Strength (psi)	26,000	Resistance (ohm/cm)	10 ¹⁴
Thermal Expansion (x 10 ⁻⁶ / °F)	5.2	Loss Factor (@ 1 Mhz)	0.01
Thermal Conductivity (BTU-in / Hr °F Ft ²)	2.8	Dielectric Constant (@ 1 Mhz)	7.5



A dense and vacuum tight, Glass Ceramic Composite that is readily machinable on conventional equipment with standard cutting tools.

No post machining heat treatments are required.

Rescor 914 Glass Ceramic is inert to oxidizing and reducing atmospheres and usable to 1000°F maximum.

Offers excellent mechanical and electrical properties and has dielectric strength of 480 volts/mil.

Rescor 914's low thermal conductivity, high impact and mechanical strength make it an ideal high temperature material.

Ready-made insulated thermocouples and Heater: Etched foil Kapton insulated heater plate



“TT” PFA
insulation



Thermocouple:

Omega 5TC type J

5TC-TT-(*)-40-(**)	40	0.08 (0.003")	PFA
--------------------	----	---------------	-----

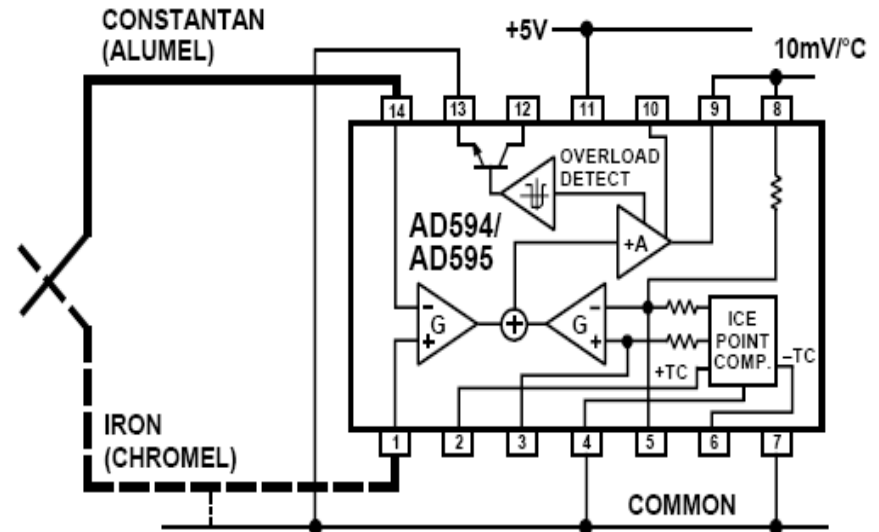
Etched foil Kapton insulated heater plate

Size in inches	Size in mm	Resistances (Ohms)	Effective area (sq in)
XY XY 0.25 1.00	6.35 25.4	25	0.18

Thermocouple interface: AD594

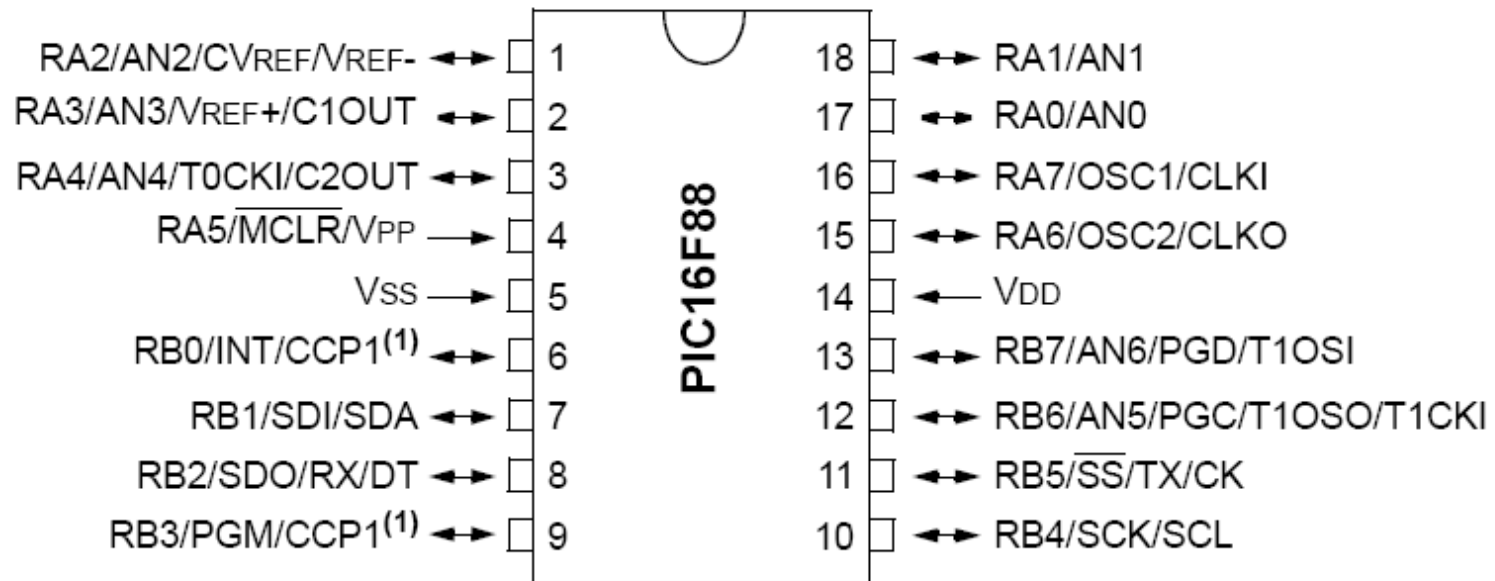
FEATURES

- Pretrimmed for Type J (AD594) or Type K (AD595) Thermocouples
- Can Be Used with Type T Thermocouple Inputs
- Low Impedance Voltage Output: 10 mV/8C
- Built-In Ice Point Compensation
- Wide Power Supply Range: +5 V to 615 V
- Low Power: <1 mW typical
- Thermocouple Failure Alarm
- Laser Wafer Trimmed to 18C Calibration Accuracy
- Setpoint Mode Operation
- Self-Contained Celsius Thermometer Operation
- High Impedance Differential Input
- Side-Brazed DIP or Low Cost Cerdip



PIC16F88

18-Pin PDIP, SOIC



PIC16F88 pinout description

Pin Name	PDIP/ SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RA0/AN0 RA0 AN0	17	19	23	I/O I	TTL Analog	PORTA is a bidirectional I/O port. Bidirectional I/O pin. Analog input channel 0.
RA1/AN1 RA1 AN1	18	20	24	I/O I	TTL Analog	Bidirectional I/O pin. Analog input channel 1.
RA2/AN2/CVREF/VREF- RA2 AN2 CVREF VREF ⁽⁴⁾	1	1	26	I/O I O I	TTL Analog O Analog	Bidirectional I/O pin. Analog input channel 2. Comparator VREF output. A/D reference voltage (Low) input.
RA3/AN3/VREF+/C1OUT RA3 AN3 VREF ⁽⁴⁾ C1OUT	2	2	27	I/O I I O	TTL Analog Analog O	Bidirectional I/O pin. Analog input channel 3. A/D reference voltage (High) input. Comparator 1 output.
RA4/AN4/T0CKI/C2OUT RA4 AN4 ⁽⁴⁾ T0CKI C2OUT	3	3	28	I/O I I O	ST Analog ST O	Bidirectional I/O pin. Analog input channel 4. Clock input to the TMR0 timer/counter. Comparator 2 output.
RA5/MCLR/Vpp RA5 MCLR Vpp	4	4	1	I I P	ST ST -	Input pin. Master Clear (Reset). Input/programming voltage input. This pin is an active-low Reset to the device. Programming voltage input.
RA6/OSC2/CLKO RA6 OSC2 CLKO	15	17	20	I/O O O	ST - -	Bidirectional I/O pin. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, this pin outputs CLKO signal which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA7/OSC1/CLKI RA7 OSC1 CLKI	16	18	21	I/O I I	ST ST/CMOS ⁽³⁾ -	Bidirectional I/O pin. Oscillator crystal input. External clock source input.

Legend: I = Input O = Output I/O = Input/Output P = Power
 - = Not used TTL = TTL Input ST = Schmitt Trigger Input

TABLE 1-2: PIC16F87/88 PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP/ SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT/CCP1 ⁽⁵⁾ RB0 INT CCP1	6	7	7	I/O I I/O	TTL ST ⁽¹⁾ ST	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Bidirectional I/O pin. External interrupt pin. Capture Input, Compare output, PWM output.
RB1/SDI/SDA RB1 SDI SDA	7	8	8	I/O I I/O	TTL ST ST	Bidirectional I/O pin. SPI Data in. I ² C Data.
RB2/SDO/RX/DT RB2 SDO RX DT	8	9	9	I/O O I I/O	TTL ST	Bidirectional I/O pin. SPI Data out. USART asynchronous receive. USART synchronous detect.
RB3/PGM/CCP1 ⁽⁵⁾ RB3 PGM CCP1	9	10	10	I/O I/O I	TTL ST ST	Bidirectional I/O pin. Low-Voltage ICSP programming enable pin. Capture Input, Compare output, PWM output.
RB4/SCK/SCL RB4 SCK SCL	10	11	12	I/O I/O I	TTL ST ST	Bidirectional I/O pin. Interrupt-on-change pin. Synchronous serial clock input/output for SPI. Synchronous serial clock input for I ² C.
RB5/SS/TX/CK RB5 SS TX CK	11	12	13	I/O I O I/O	TTL TTL	Bidirectional I/O pin. Interrupt-on-change pin. Slave select for SPI in Slave mode. USART asynchronous transmit. USART synchronous clock.
RB6/AN5/PGC/T1OSO/ T1CKI RB6 AN5 ⁽⁴⁾ PGC T1OSO T1CKI	12	13	15	I/O I I/O O I	TTL I ST ⁽²⁾ ST ST	Bidirectional I/O pin. Interrupt-on-change pin. Analog input channel 5. In-circuit debugger and programming clock pin. Timer1 oscillator output. Timer1 external clock input.
RB7/AN6/PGD/T1OSI RB7 AN6 ⁽⁴⁾ PGD T1OSI	13	14	16	I/O I I I	TTL I ST ⁽²⁾ ST	Bidirectional I/O pin. Interrupt-on-change pin. Analog input channel 6. In-circuit debugger and ICSP programming data pin. Timer1 oscillator input.
Vss	5	5, 6	3, 5	P	-	Ground reference for logic and I/O pins.
VDD	14	15, 16	17, 19	P	-	Positive supply for logic and I/O pins.

Legend: I = Input O = Output I/O = Input/Output P = Power
 - = Not used TTL = TTL Input ST = Schmitt Trigger Input

- Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
 4: PIC16F88 devices only.
 5: The CCP1 pin is determined by CCPMX in Configuration Word 1 register.

PICBasic Pro Programming

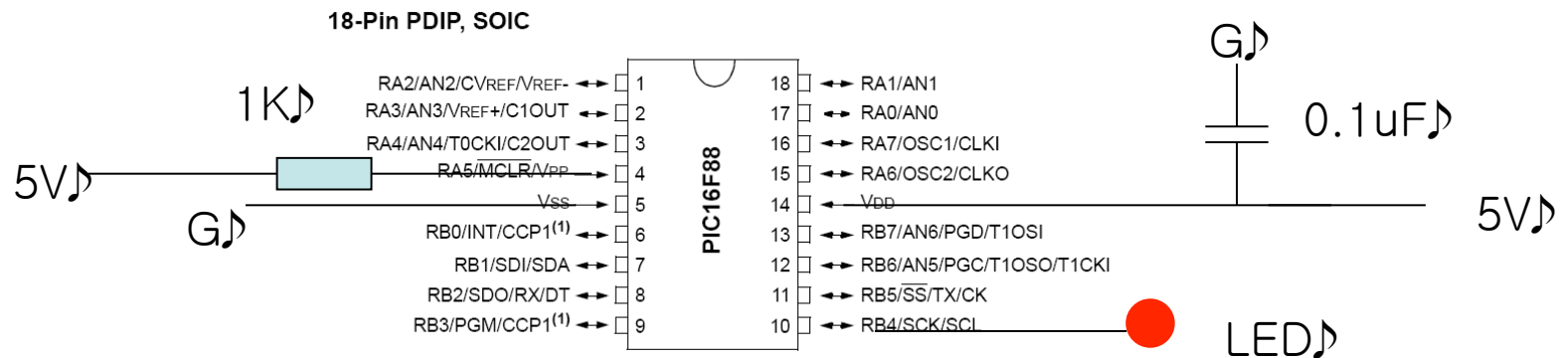
- PicBasic Pro is a compiler that uses a pseudocode approach to translate user friendly BASIC code into assembly language code that is created in a separate *.asm file.
- The assembly code is then compiled into hexadecimal machine code (*.hex file) that the PIC can interpret.
- The hex code file is then downloaded to the PIC and remains stored semi-permanently in EEPROM even when it is powered off. It will remain in PIC memory until it is erased or overwritten using the DEVELOPMENT PROGRAMMER.

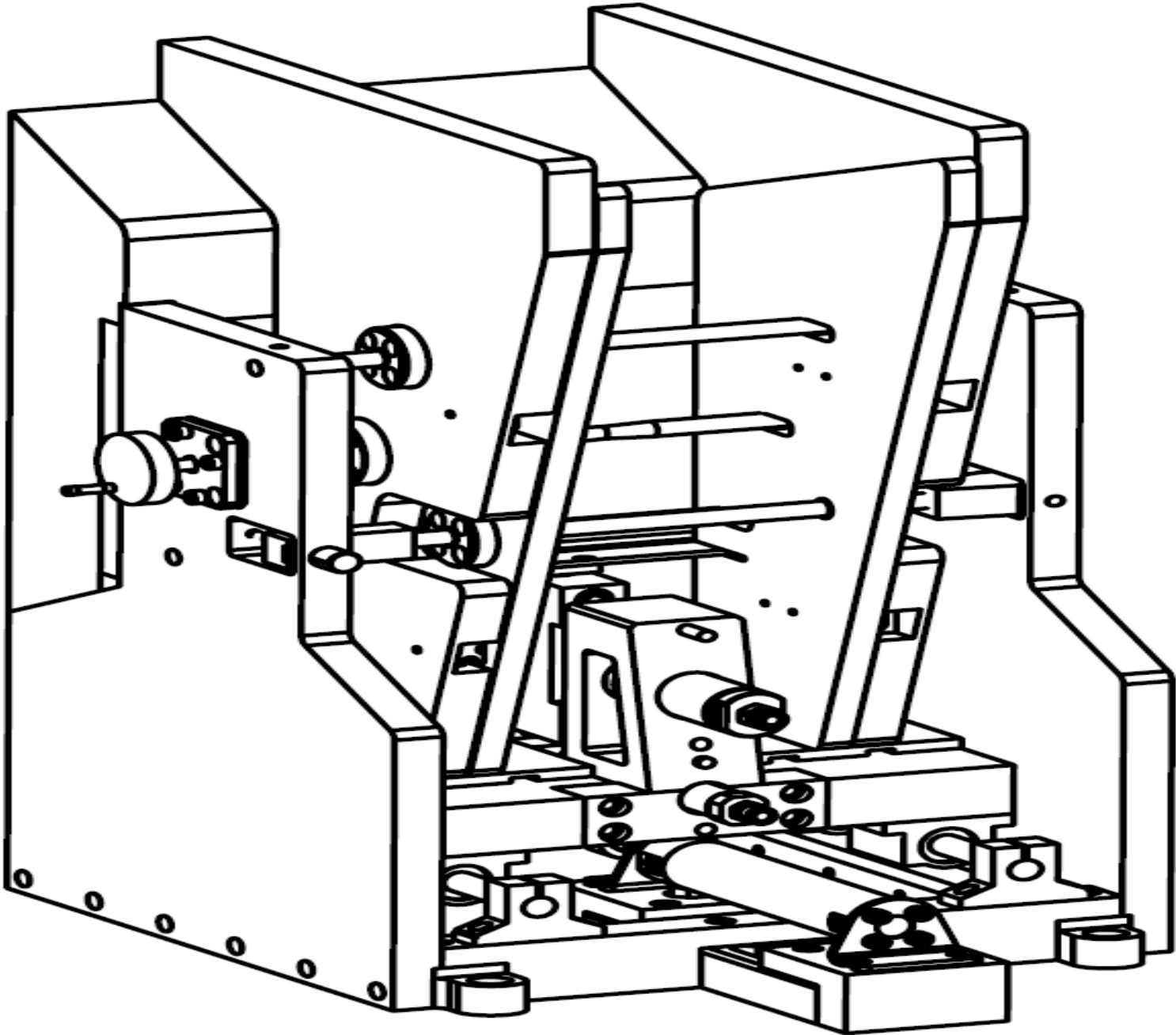
Blink an LED

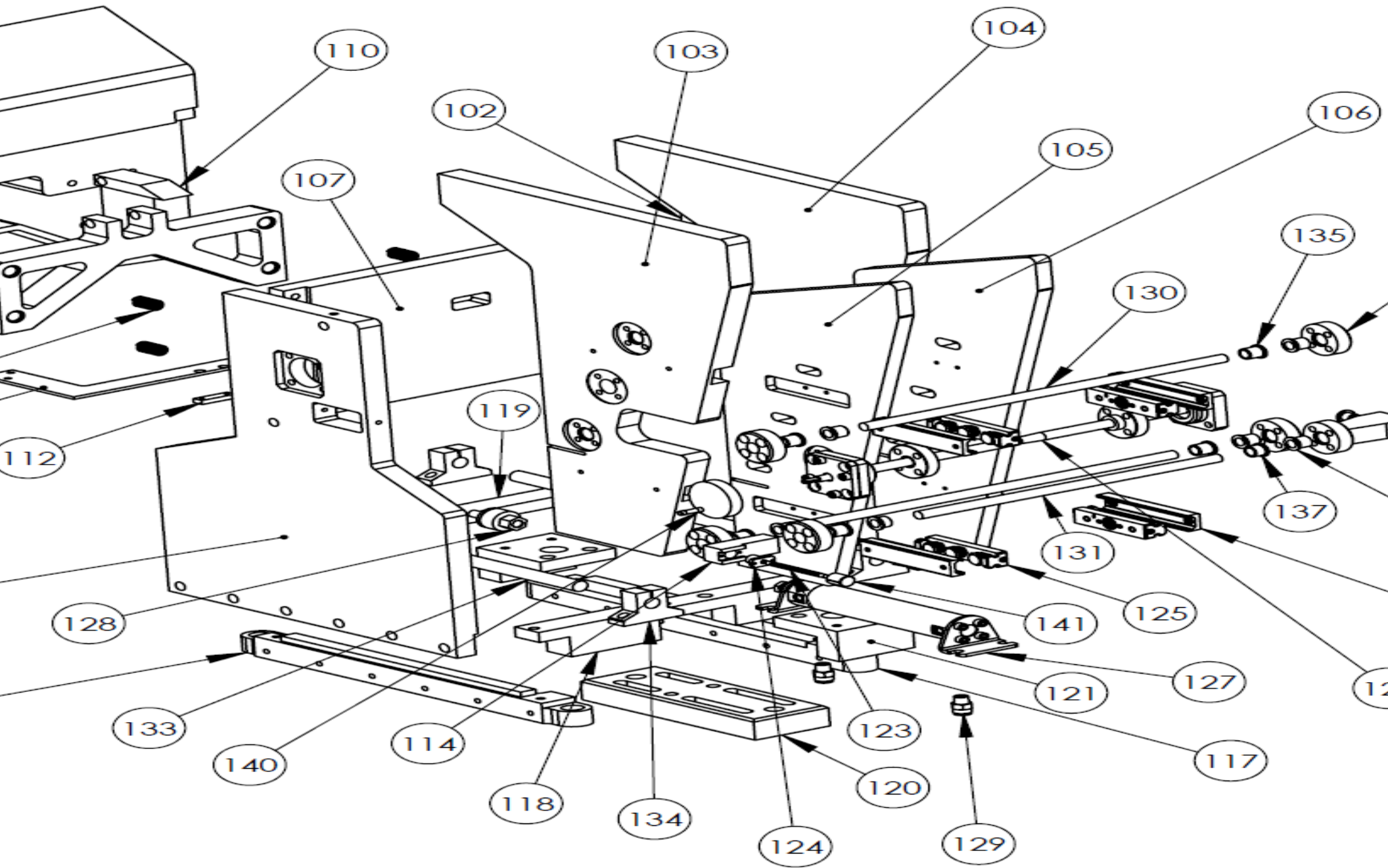
Build an LED blinking circuit using a PIC16F88.

```
`lab6a.bas
`Example program from to blink an LED connected to PORTB.0
`Configure the internal 8MHz internal oscillator
define osc 8
osccon.4=1 : osccon.5=1 : osccon.6=1
loop:      High PORTB.4          ' Turn on LED connected to PORTB.4
           Pause 500              ' Delay for .5 seconds
           Low PORTB.4           ' Turn off LED connected to PORTB.4
           Pause 500              ' Delay for .5 seconds
           Goto loop             ' Go back to loop and blink LED forever
End
```

Modify the program so that the LED can blink twice faster! -> Report item.





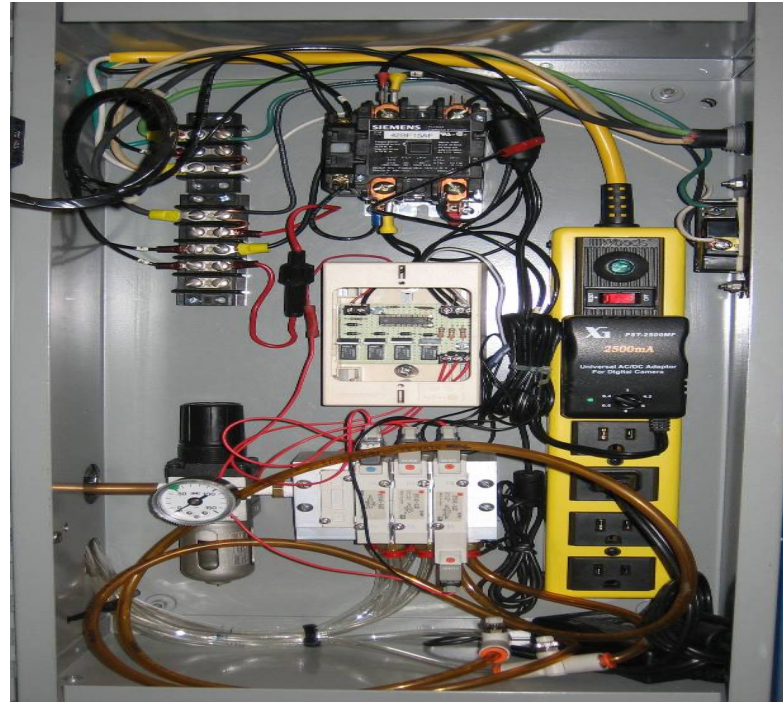


CONTROL BOX - FRONT



PART NUMBER	TITLE	DESCRIPTION	QTY
1	CONTROL BOX	GREY, STEEL	1
2	EMERGENCY STOP	PULL TO RESET	1
3	ON/OFF BUTTON	MCMaster-CARR: 8382K232	1
4	START BUTTON	MCMaster-CARR: 8382K211	1
5	STOP BUTTON	MCMaster-CARR: 8382K221	1
6	ON INDICATOR	MCMaster-CARR: 5339T143	1
7	START INDICATOR	MCMaster-CARR: 5339T142	1
8	LCD COUNTER	MCMaster-CARR: 1737T52	1
SEE APPENDIX A FOR SPECIFICATION SHEETS			

CONTROL BOX-INSIDE



Arduino

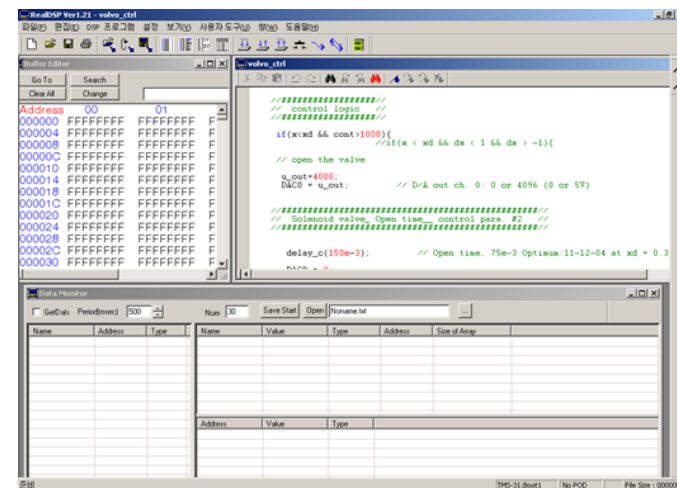
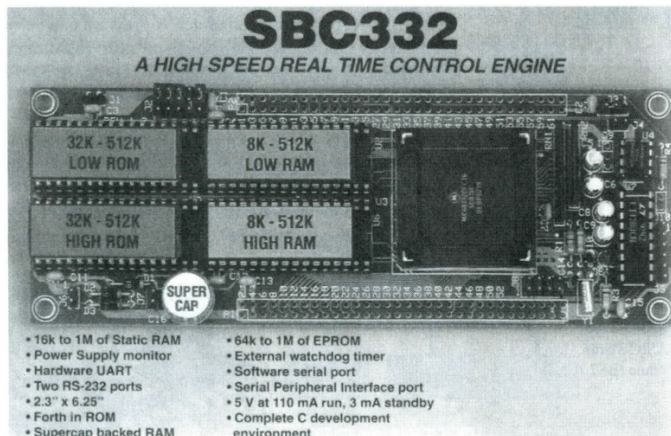
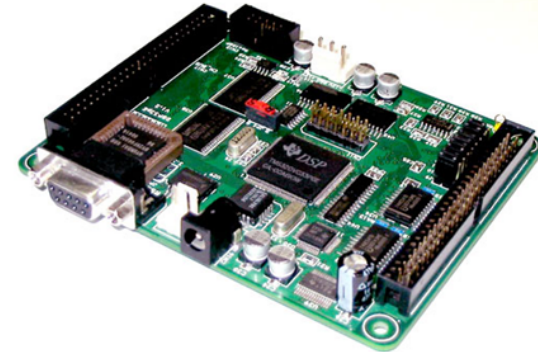
Arduino is a popular “open source” single board microcontroller. It is designed to make the process of using electronics in multidisciplinary projects more accessible.



Single-Board Computers

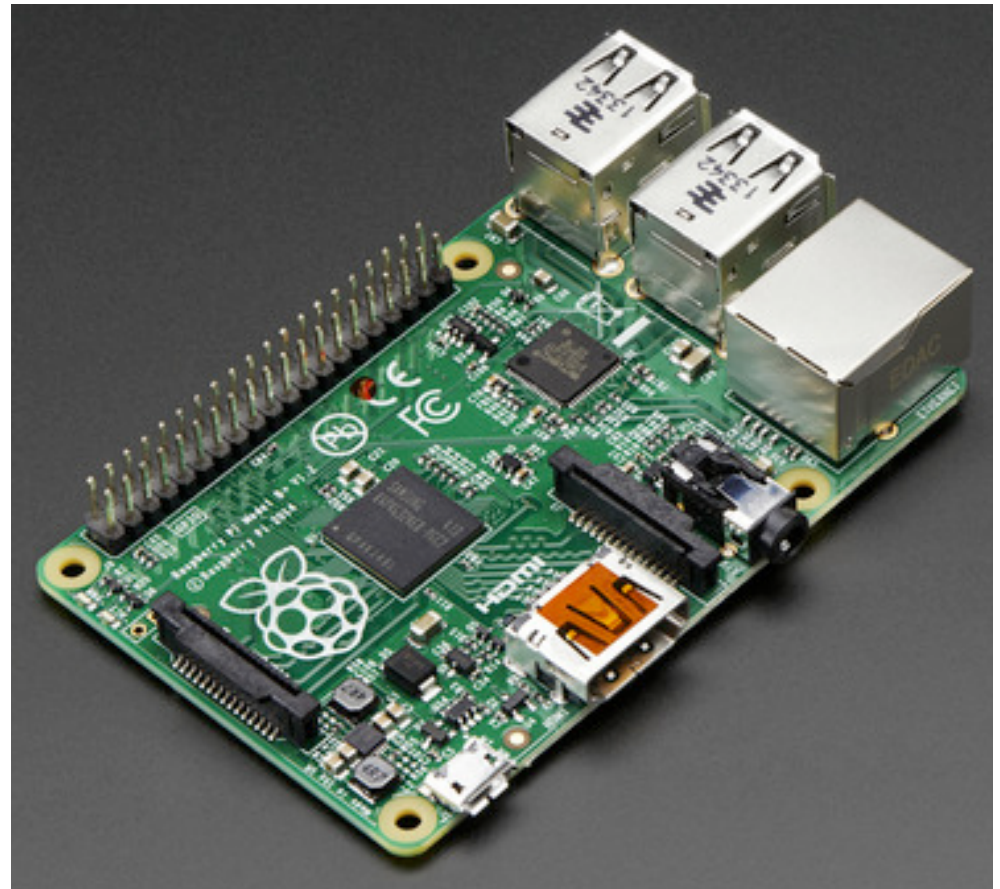
- A computer on a single board. Programmable for I/O control and the ability to use high level peripherals.

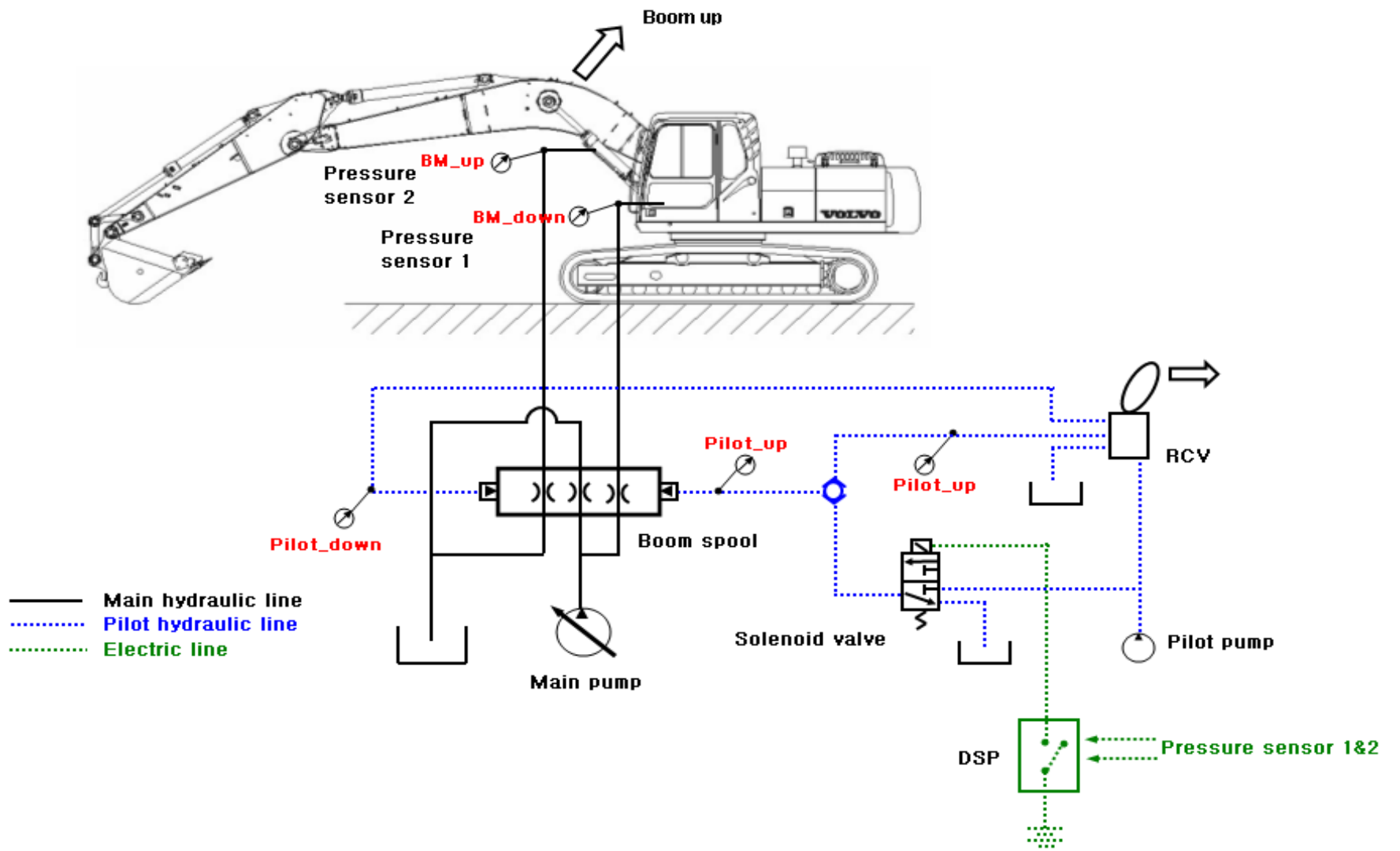
TI TMS320 chip,
Realsys-DSP33-PLUSE



Raspberry Pi

- The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard.
- It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games
- It also plays high-definition video.
- Mostly it runs a stripped down linux distribution
 - You write code similar to how you would write for Linux
 - Comes with easy to use python scripting language for programming







♪

- ▶ Pressure sensors
 - Boom-up chamber
 - Boom-low chamber
- ▶ Boom stroke sensor

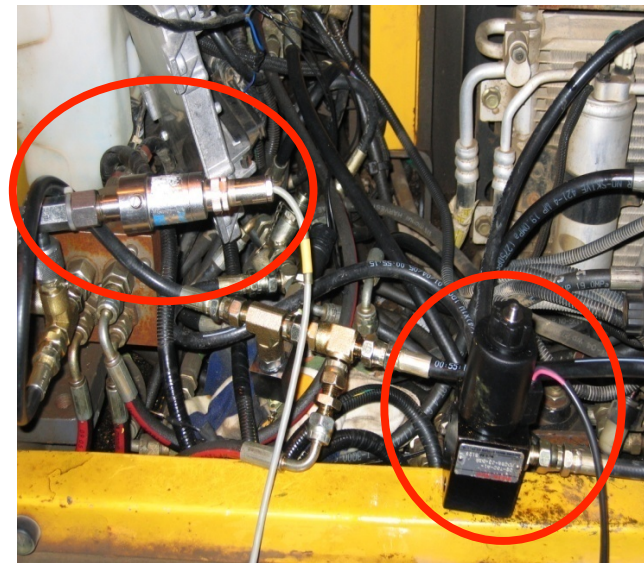
♪



♪

- ▶ DSP controller

♪



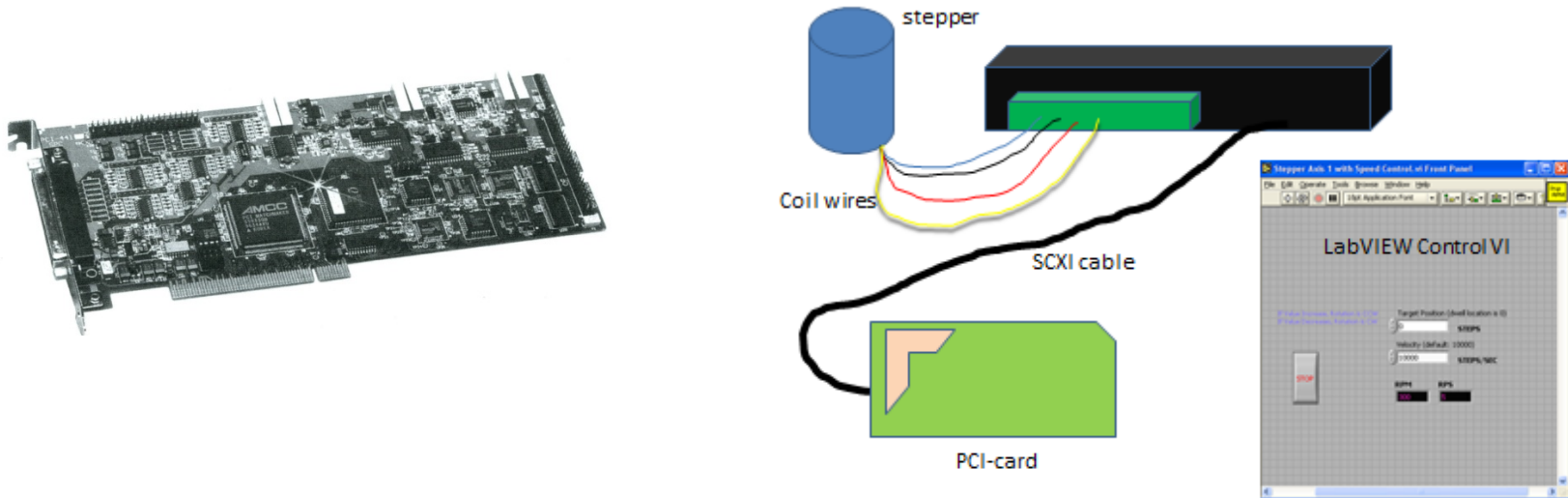
- ▶ Solenoid actuation control led by DSP

♪

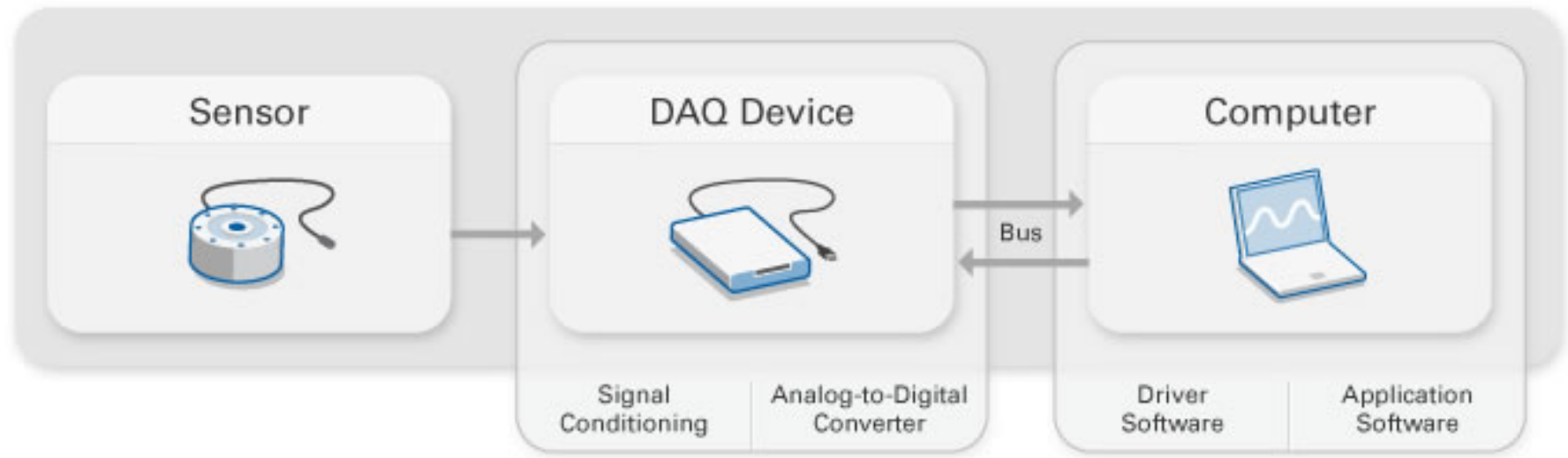
Personal Computers

- PCs with dedicated I/O and data acquisition cards and specialized software may be used as controllers.

Demo Application 2



NI board controller



Example robot finger demonstration

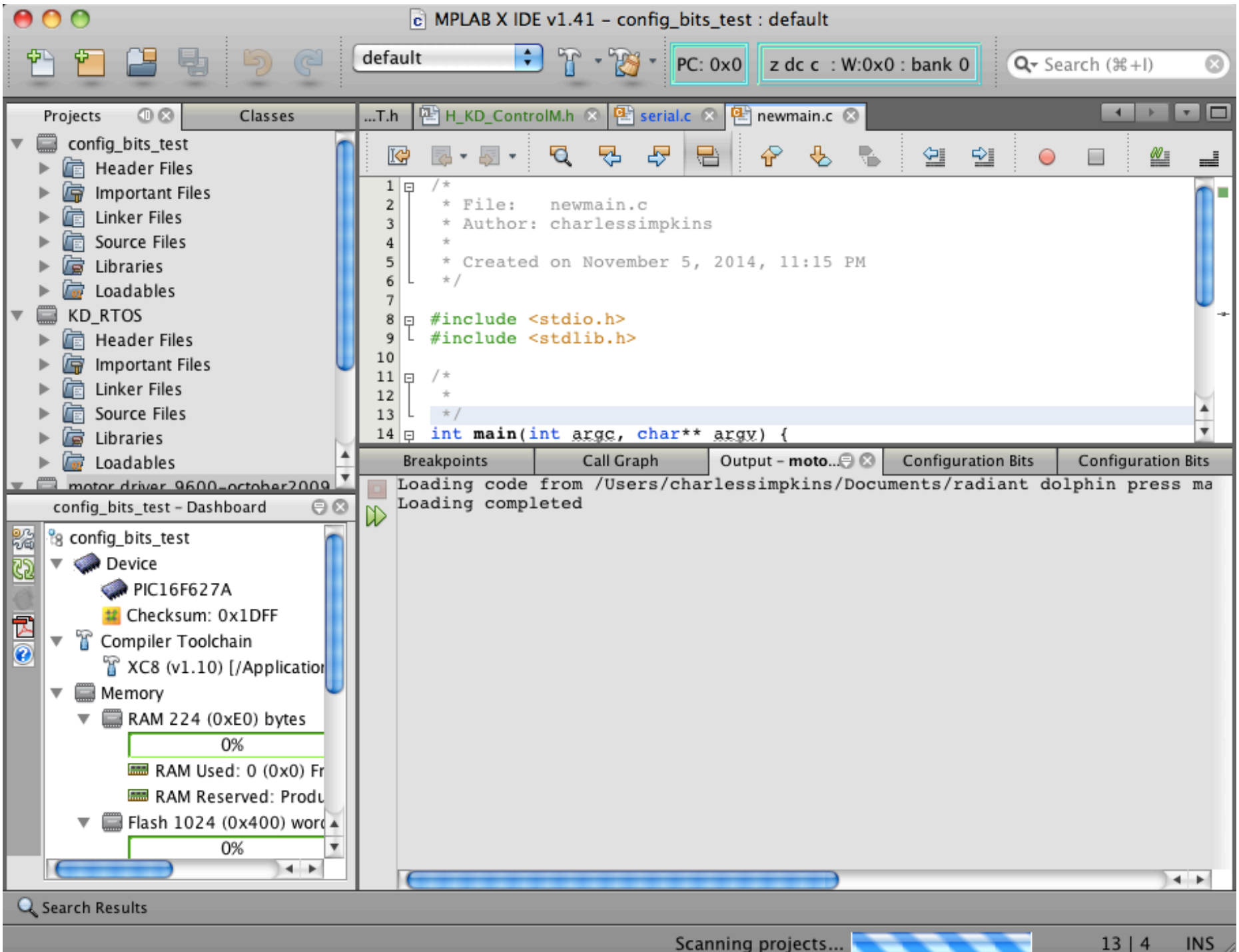
Programming languages for microcontrollers

- Processors are programmed with a machine language hex file
- It isn't convenient for humans to read this
- That's why higher level languages are created
 - C, C++, Basic, Assembly (lower level), Pascal, Fortran, etc
 - Makes it simpler for humans to generate instructions for the processors to follow

```
:10007000DACFE6FFE1CFD9FFE2CFDAFFFD0EDBCF2B
:1000800001F0800E01140BE0FC0EDBCF00F0FD0E42
:10009000DBC0F1F0006C011E000E012206D0FC0E29
:1000A000DBC0F0F0FD0EDBCF01F000C0F3FF01C09D
:1000B000F4FFE552E5CFDAFFE7CFD9FF1200D9CF41
:1000C000E6FFDACFE6FFE1CFD9FFE2CFDAFFE1507A
:1000D000FA0FE96EFF0EE220EA6ED890030EEB50A5
:1000E000000A08E6000EEE5C000EEE58000EEE5818
:1000F000000EEE5815E2FA0EDBCF00F0FB0EDBCF60
:1001000001F0FC0EDBCF02F0FD0EDBCF03F0000EA2
:10011000006C011E0122021E0222031E03220CD0CB
:10012000FA0EDBCF00F0FB0EDBCF01F0FC0EDBCFD5
:1001300002F0FD0EDBCF03F000C02EF001C02FF067
:1001400002C030F003C031F0E552E5CFDAFFE7CF6F
:10015000D9FF1200D9CFE6FFDACFE6FFE1CFD9FF12
```

Compilers and programming environments

- You write code in a development environment
 - picBasic Pro
 - MPLAB X
 - gcc, etc



Programming the chip

- The code is then compiled and downloaded in to the processor
- This is done either through a “programmer,” or using a bootloader
 - Bootloaders work for certain types of processors that are capable of ‘self-programming’
 - Small pieces of code that instruct the processor to shift the rest of the program into memory
 - Arduinos typically use this method

Debugging

- Allows you to step through code and look at variables, execution paths, and data
- Extremely useful and necessary for complex code