

ARDUINO PROGRAMMING & ITS APPLICATIONS

Day-One

Zuhaib A. Shaikh, Asst. Prof., CSE Deptt.,QUEST Web: <u>zuhaib-shaikh.neocities.org</u>

Outline

Day	Activity
1	 i. Introduction to Arduino development board and Arduino IDE ii. C/C++ language overview iii. Basic input / output with Arduino iv. Overview of Proteus simulation software for Arduino simulation
2	i. Interfacing and glowing LEDs with different patternii. Interfacing push button and piezo buzzeriii. Interfacing a temperature sensor with Arduino
3	i. Familiarization with Serial Monitor for input and for outputii. Interfacing LDR sensor with Arduinoiii. Interfacing PIR motion sensor with Arduino
4	i. Interfacing Arduino with LCD (16x2), relay and Servo motorii. Interface Arduino with Sonic Sensor for obstacle detection
5	i. Interfacing shift register and 7-segment display with Arduinoii. Interfacing HC-05 Bluetooth module with Arduinoiii. Driving GSM modem with Arduino
Prerec - Kı - Kı	uisites: nowledge of C++ nowledge of basic electronic components

Introduction

- Arduino (a Italian company) is opensource HW/SW company
 - Designs development board for microcontrollers and microcontroller kits
 - Its IDE supports C/C++ like language for writing and burning the code (HEX file) into Arduino Board
- Useful for learning, experiments and prototyping
- Different Models with different microcontrollers:
 - Arduino Uno
 - Arduino Mega
 - Arduino Nano
 - Arduino Due, etc,
- Selection of particular model depends on requirements
- Many shield and sensors are compatible with Arduino

Arduino Uno

- Basic board, suitable for most of the projects and mostly supported by Arduino shields
- It has:
 - ATmega Microcontroller
 - Digital I/O pins (0 to 13)
 - Pins with tilde (~) can generated PWM analog signals
 - Power pins
 - Analog pins (A0 to A5)
 - Control Pins
 - Power connector
 - USB Port (for connecting with PC)
 - LEDs (Power, TX, RX & L-for pin 13)
- *I2C Inter-integrated Component for Master-Slave Communication
 *ICSP – In-Circuit Serial Programming
- 7 to 12VDC input 2.1mm x 5.5mm USB-B port Male center positive to computer Reset button ICSP for **USB** interface (I2C) SCL - Serial clock (I2C) SDA - Serial data Pin-13 LED Not connected (SPI) SCK - Serial clock I/O Reference voltage IOREF (SPI) MISO - Master-in, slave-out 12 Reset RESE (SPI) MOSI - Master-out, slave-in ~11 3.3V Output ~10 (SPI) SS - Slave select 5V Output ~9 Ground Ground Note: Pins denoted with "~" Input voltage are PWM supported Analog pin 0 A0 Analog pin 1 Analog pin 2 Interrupt 1 Analog pin 3 Interrupt 2 (I2C) SDA TXD $TX \rightarrow 1$ (I2C) SCL ax 4 RXD ATmega328 microcontroller IC VCC RESET MOSI SCK ICSP for GND ATmega328 MISO

Voltage

regulator

16MHz

crystal

ATmega16U2

microcontroller IC/USB controller

Arduino IDE

- Allows to write C++ like programs (sketches), compile those and burn those on Arduino board Sketch Arduino Serial
- IDE is divided into
 - Command Area
 - Title Bar
 - Menu Items
 - Icons
 - Text Area
 - Message Window
- Port and Board
 - IDE supports many Arduino boards
 - Connected as serial COM port
 - Port and Board should be selected before working on project
- Serial Monitor
 - IDE allows to input/output from/to PC & Arduino using IDE



Arduino C++

- Arduino C++ is based on traditional C++ except standard library
- Similar keywords, control structures, primary and secondary datatypes etc.
- Arduino library
 - Setup function
 - The function which is to be executed first for only once
 - This function is usually used to set certain things like Digital I/O pins, Analog pins, baud rate etc.
 - Loop function
 - The function which is to be executed repeatedly
 - All instructions are written inside loop function which are meant to be executed infinite times
 - Basic I/O functions
 - **pinMode**(*pin_no, mode*) sets the digital pin in **INPUT** or **OUTPUT** mode (mostly in setup function)
 - **digitalWrite**(*pin_no/reference*, *value*) assigns HIGH or LOW value to particular pin number or its reference until interrupted
 - Variable = digitalRead(pin_no/reference) Reads digital value from particular pin number or its reference to defined variable
 - Serial.begin(baud_rate) To set serial baud rate for serial monitor
 - Serial.print(string_and/or_variable) and serial.println(string_and/or_variable) prints string (enclosed in double quotation) and/or variable value on serial monitor without and with new line respectively
 - delay(milli_seconds) describe microcontroller to do nothing (wait state) in specific time defined in milli-seconds

Example 1: LED Blink

Circuit



Code

* LED L is connected to pin number 13 for testing purpose by default

Verifying and uploading sketch

- Once code is written, it is necessary to compile (verify) the code to generate HEX file
- If code is without errors, its hex file will be temporary stored on PC by displaying following information:
 - Size of the program (in bytes)
 - Size of total Arduino memory (in bytes)
- HEX (output) file can be uploaded on Arduino using Upload button
 - TX & RX LEDs will blink during upload
 - Arduino board will start executing HEX file once upload is done
- If the sketch has syntax errors
 - Display error message by mentioning the exact or expected error
 - Compilation and uploading can't be continued until errors are removed





Simulating on Proteus

- The example can also be simulated on Proteus simulation software rather than physical implantation
- ISIS(Intelligent Schematic Input System), a tool of Proteus allows users to design and simulate electronic circuits easily



- ISIS supports wide variety of components
- Selected components are visible in object select window, while complete circuit is in schematic edit space
- Components (including power source) can be selected via menu buttons

Simulating on Proteus

The components from **P**ick device window can be search via their names or model numbers 0

CDF

Boot sheet 1

E Pick Devices ie, no de

Vacano Pratet Wards Words?

Show only parts with models? Datequi,

 $\leftrightarrow \pm$

n Greph Source Debug Librery Templete System Help

Becute (D):

Deves.

PLAY

•

STEP

PAUSE

noticer: view AFEUING MEGATOC ADDUINO Arduino MEGA (ATmega ICO) AFEUING MEGA2060 F3 (ARDUINO Arduino MEGA2060 F3+3

alegnik Suli is legniy Taxini dan

: ANDUING UNG N) : ANDUINGL D : Verday, Natsh (1, 2013 a) 11:12 24 PM - multin : Althesa Anton (Ibin = 3

PIC18F462

No Message:

-1850.0

9

ATOUND LIND FOR Providen

VS- DLL -oce WF2DLLI, 4114

STOP

0

C UNITILED - 1313 Protession	al				5" X
File Were Ecit Tools	Pick Devices				
	uwords:	Results (128)			FIC18F
	LISF SCATCH IIC	PIC18F2680	Library PICMICRO	Vescription A FIC18 Microcontroller (64kB code, 33288 data, 10248 EE	VSN I
Sh	orv only parts with models?	PIC18F2682 PIC10F2685	PICMICRE	FIC18 Microcontroller (80kB code, 33288 data, 1024B EE FIC10 Microcontroller (95kB code, 33268 data, 1024B EE	
IA M	Il Categores) icroprocessor ICs	PIC18F26K20 PIC18F26K22 PIC18F4220	PICMICRO	FIL18 Microcontroller (64kB code 3336B data, 10248 EPF FIC18 Microcontroller (64kB code, 3896B data, 256E EEF FIC18 Microcontroller (4kB code 512B data, 2568 EPFON	
		PIC18F4221 PIC18F4320	PICMICRO	FIC18 Microcontroller (448 code, 5128 data, 2568 EPROI FIC18 Microcontroller (848 code 5128 data, 2568 EPROI EC10 Microcontroller (848 code 5128 data, 2568 EPROI	
CAPACITCR PILT8F452	D 11	PIC10F4001 PIC10F4001 PIC18F43K20	PICMICRU	FIL 18 Microcontroller (3k8 code, 5126 cata, 2569 EPHU) FIC10 Microcontroller (0k8 code 7600 data, 2569 EPHU) FIL18 Microcontroller (8k8 code 5128 data, 2568 EPHU)	
RESISTOR	Double	PIC18F43K22 PIC18F4410	PICMICRO	FIC18 Microcontroller (9x8 code, 5128 data, 2568 EEPF(FIC18 Microcontroller (15k8 code 7668 data, 06 EPF0M	
	click	PIC18F4420 PIC18F4423	PILICRO	FIC18 Microcontroller (15kB code, 756B data, 256B EEPF FIC18 Microcontroller (15kB code 7688 data, 256B EPR0 FIC18 Microcontroller (15kB code, 766B data, 256B EPR)	
	to appear	PIC10F4401 PIC18F4439	PIC ICRO	FIC10 Microcontroller (15kD code 7600 dete, 2568 EFF10 FIC18 Microcontroller (12kB code 6408 data, 2568 EFF10 =	FLB PT
	here	PIC18F4455 PIC18F4455 PIC18F4458	PIC CRC PIC CRC	FIC18 Microcontroller (19kB code 20488 data, PorcAre, 1 FIC18 Microcontroller (24kB code 20488 data, 256B EEPI FIC18 Microcontroller (24kB code 20498 data, 256B EEPI	
		PIC18F448 PIC18F4480	PIC CRO PIC CRO	FIC18 Microcontroller (15kB code, 756B data, 256B EEPF FIC18 Microcontroller (15kB code, 756B data, 255B EEPF	
20		PIC18F44J10 PIC18F44K20 PIC18F44K22	PIC CRC PIC CRC	FIL18 Microcontroller (15kB code 1024B date, US EPFID) FIL18 Microcontroller (15kB code 7688 date, 256B EPFID) FIL18 Microcontroller (15kB code, 768B date, 256B EEPF	
		PIC18F4510 PIC18F4515	PICK RO PICMICRO	FIC18 Microcontroller (32kB code 1536B data, 03 EPFON FIC18 Microcontroller (43kB code 3969B data, 03 EPFON	
		PIC18F452 PIC18F4520 PIC18F4523	PICMICRO	FIC18 Microcontroller (32kB code, 15368 data, 2568 EEF FIC18 Microcontroller (32kB code 15368 data, 2568 EPR FIC18 Microcontroller (32kB code, 15368 data, 2568 EPF	
CON ↔ T Root sheet 1 Ma	nufacture:	PIC18F4525 PIC18F4539	PICMICRU	FIC18 Microcontroller (43kB code 39638 data, 2568 EPR FIC18 Microcontroller (24kB code 1408B data, 2568 EPR	

- The simulation can be started, paused and stopped using the buttons at bottom of ISIS Ο
- ISIS-Proteus 7 doesn't not provide Arduino Library by default, therefore: 0
 - Download Arduino Library and extract it 0

https://www.theengineeringprojects.com/ArduinoProjects/Arduino%20Library%20for%20Proteus.rar

Copy the extracted files into (ISIS is needed to be closed): 0

C:\Program Files (x86)\Labcenter Electronics\Proteus 7 Professional\LIBRARY

After including the Arduino library, Arduino board can be added pick device window by Ο writing name of the board.

Simulating on Proteus

- Arduino module now needs the HEX file for simulation.
- HEX file can be obtained via Arduino IDE
 - Open preference window from File tab in IDE
 - Under setting tab, apply locate "Show verbose output during..." and check the compilation box



- After verifying the sketch, HEX file can be located from the path shown in message window
- File path can be viewed by pasting the path in Run dialog box
- HEX file is then to be included in Proteus by double clicking on Arduino & selecting HEX file
- Simulation can be started after completing the circuit and adding HEX File

Example 2: LED Fade

- Connect LED with PWM pin (e.g. pin 9) and with GND
- The example code can be opened from *File>Examples>01.Basics>Fade*



Example 3: Traffic Signal (Exercise)

- Connect 3 LEDs with digital pins (e.g. pin 9) using 270ohm resistors and with GND
- Write program to glow those LEDs with different time periods, i.e. RED = 3s, YELLOW = 0.5s and GREEN = 4s
- The glowing order should be like Red Yellow Green Yellow Red



Example 4: Controlling DC motor

- DC motor has 2 terminals +ve & -ve. Motor start working when those terminals are connect to DC supply.
- Components:

0	12V DC power supply (5V in case you have a 5V DC motor)	X 1
ο	1N4001 or equivalent back EMF blocking diode	X 1
0	2N2222 NPN transistor	X 1
0	5V or 12V DC motor	X 1
0	270 ohm resistor	X 1



Using Breadboard

- Allows to design circuit connections without permanently soldering those
- Divided into matrix of rows and columns
- Columns of single block are connected with each other
- While, 2-piece or 4-piece *rails* are use to be connected with power supply to provide to components



Exercise

- Simulate a ISIS-Proteus Project using Arduino with eight LEDs.
- Blink them in an endless loop in the sequence:

1-2-3-4-5-6-7-8-7-6-5-4-3-2-1-2-3-...

- The delay b/w steps should be following by three combinations
 - one second
 - Half second
 - Quarter second

Questions

