



Introduction to Microcontrollers

Shivendu Bhushan

Sonu Agarwal

Things to be covered today...

- Embedded System – Introduction, Examples
- Microcontrollers - basic features
- Input and output from a micro-controller
- Programming a micro-controller
- Interfacing Character LCD with Micro-controller
- How to use Infrared – Tsop sensor ?

Embedded Systems

- Gadgets and devices
- Self controlled devices
- Contains I/O devices, storage devices and a central 'controller'

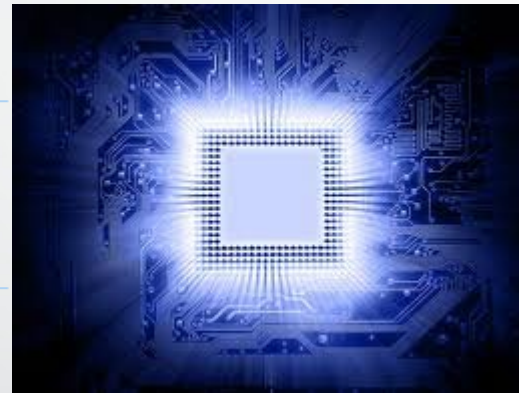
Example: Music player



Output



Input



Controller

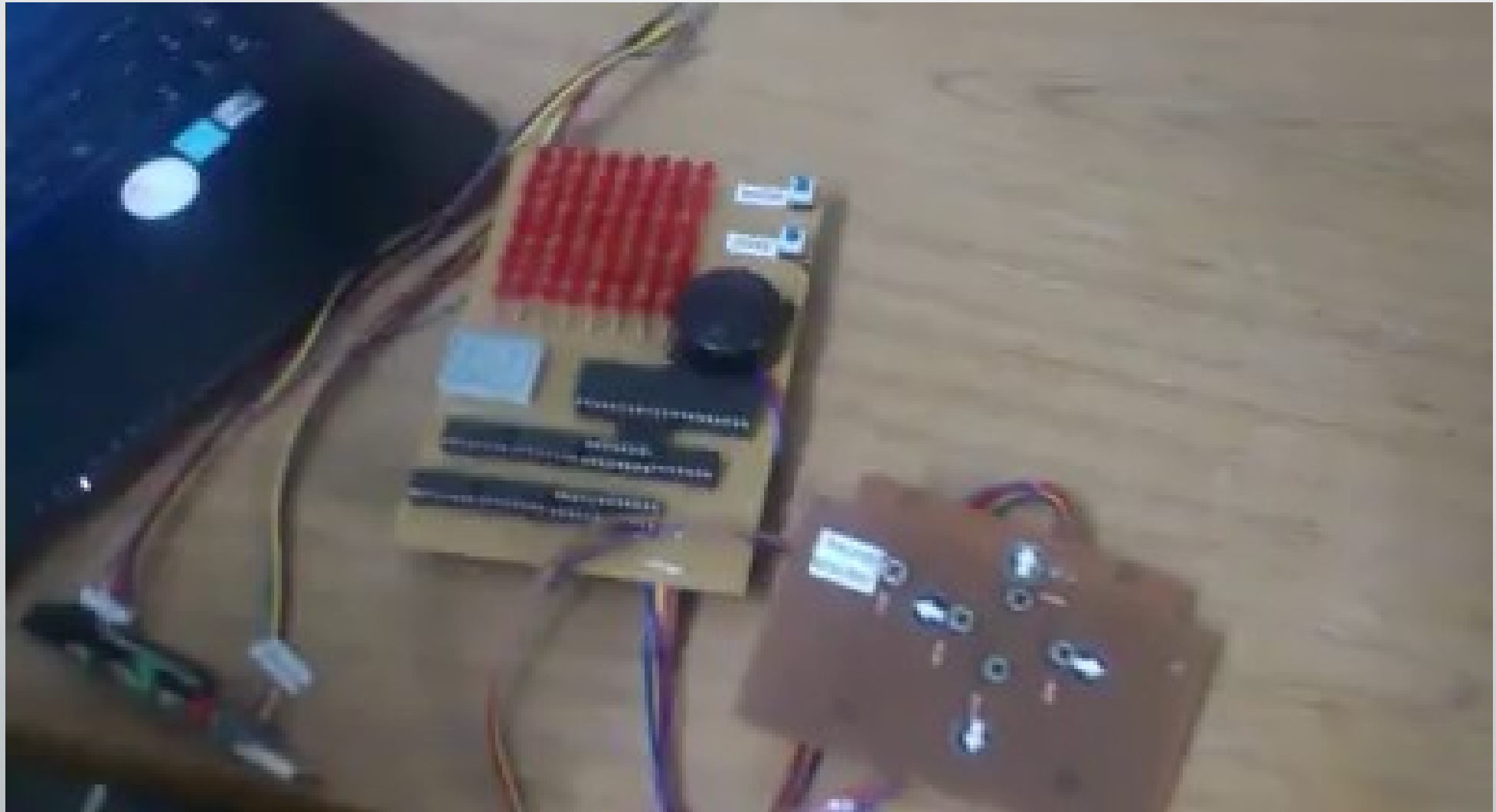


Output

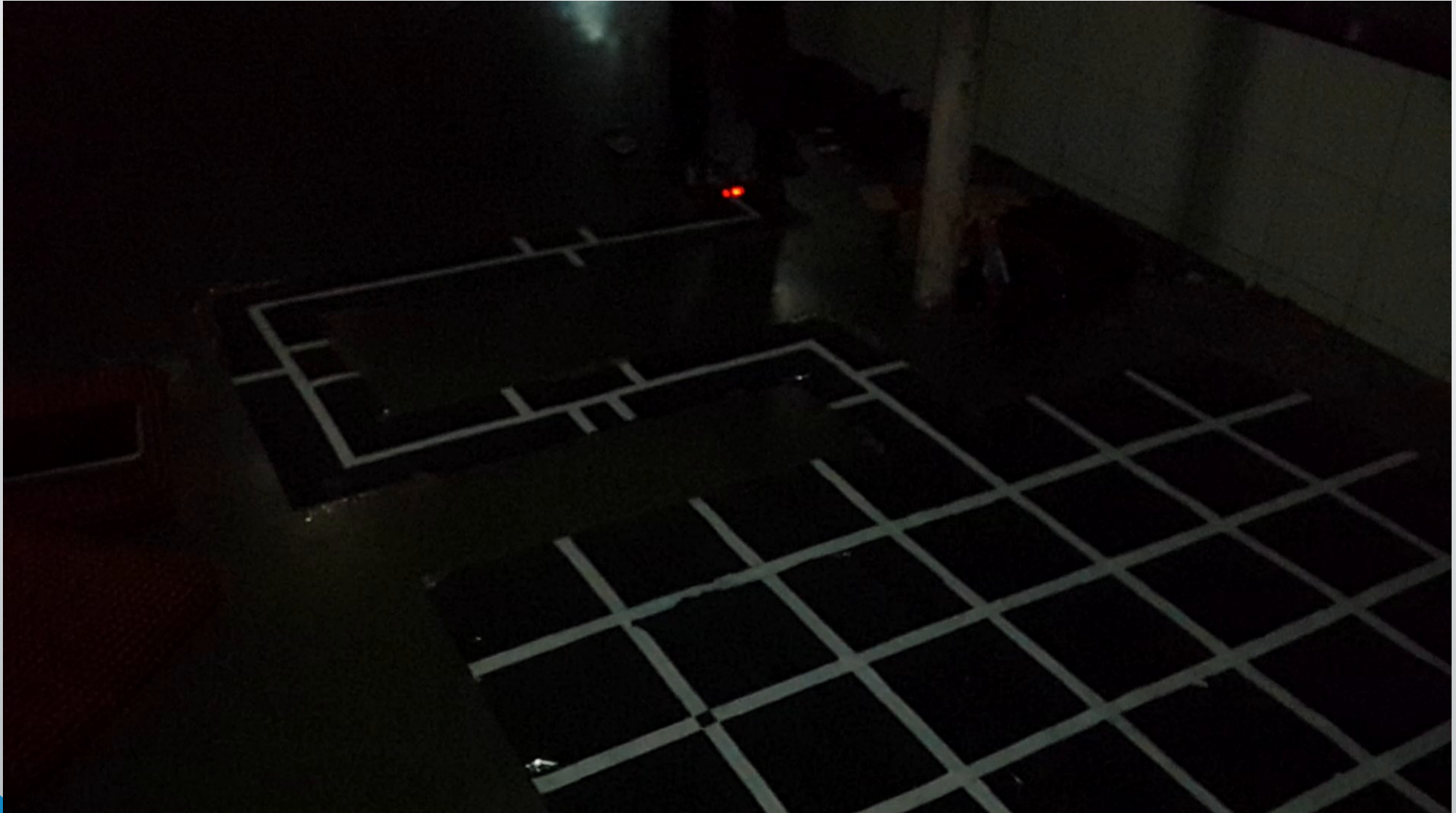


Storage Device

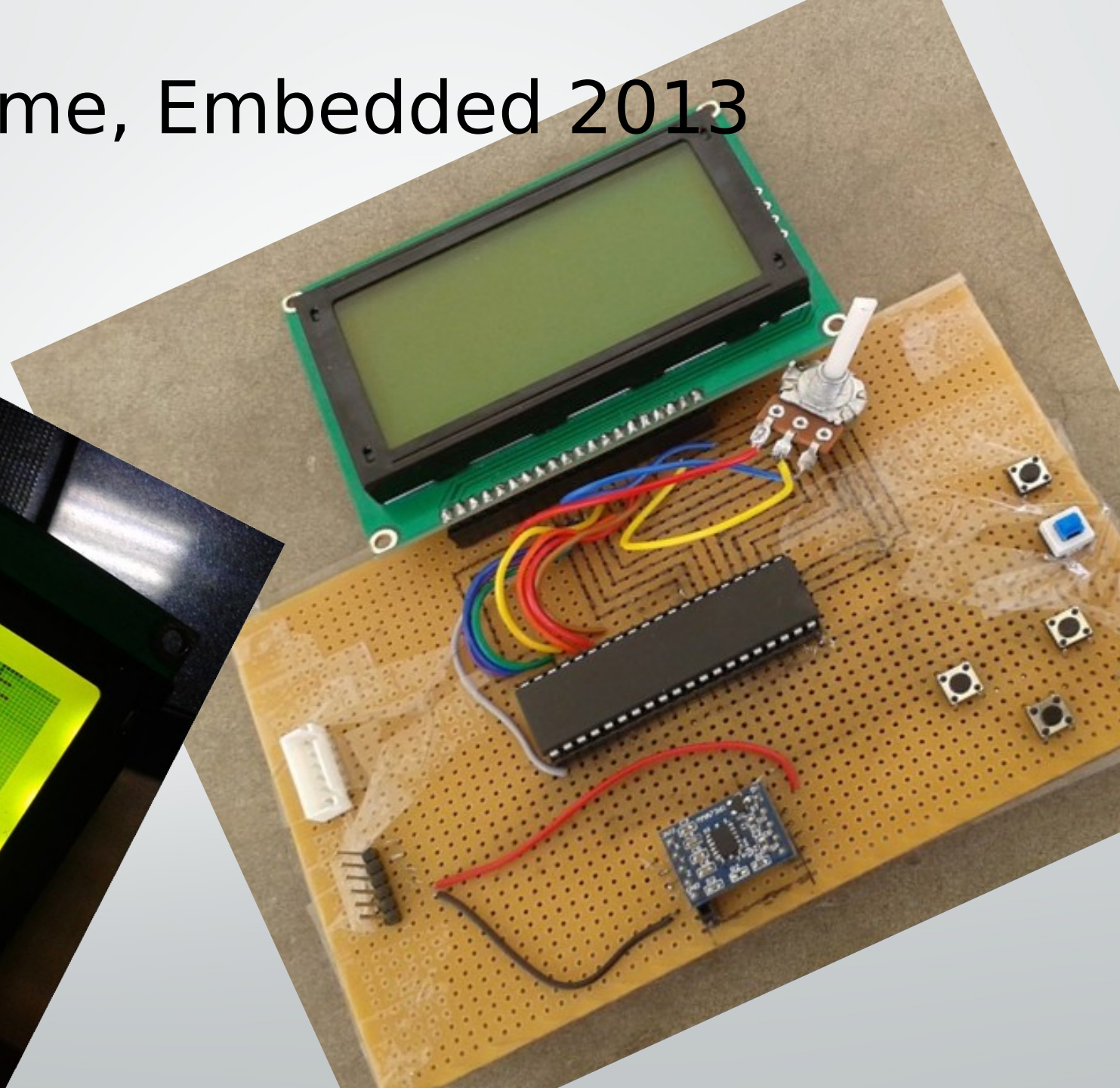
Snake Game, Electromania 2013



Line Following Bot, Techfest 2014



Maze Game, Embedded 2013



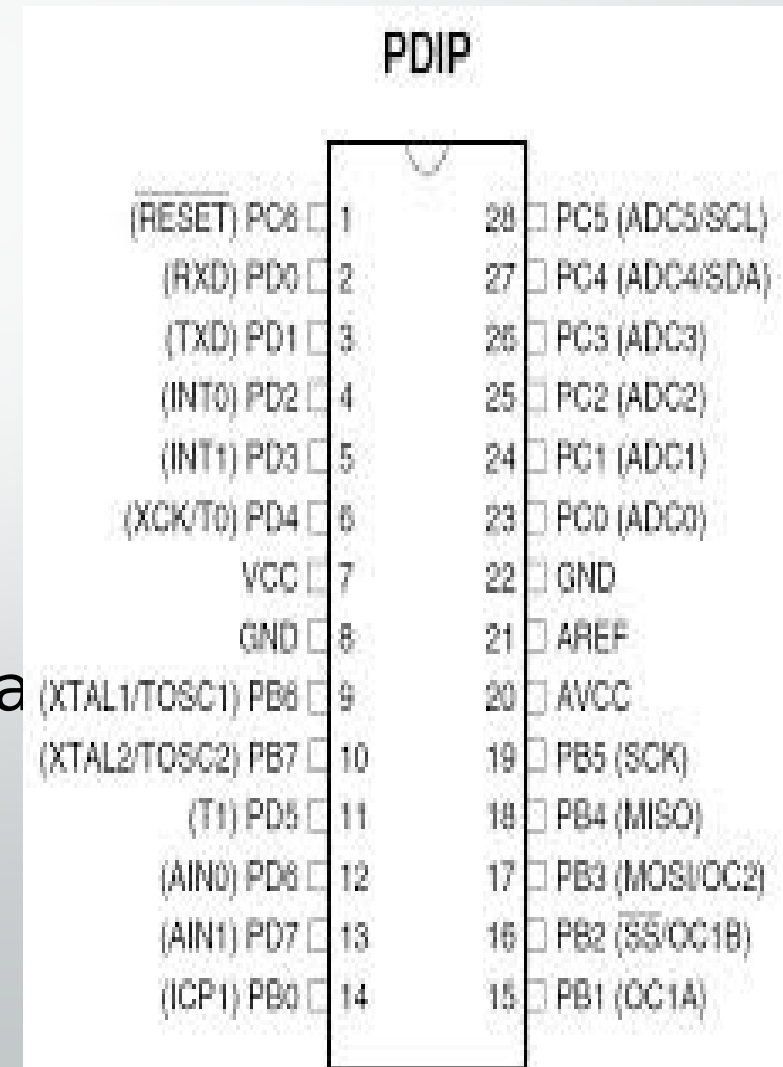


Micro-Controllers

- Why “micro”? Larger controllers are available too: processors that run computers
- Out of several available vendors like Atmel, Intel, ARM, Cypress, etc. We will use Atmel ATmega microcontrollers
- Like computers they execute programs. We will use C as the coding language

ATMEGA 8

- 28 pin IC
- 23 pins for I/O
- 5 pins reserved
- I/O pins divided into 3 groups of 8* pins, called ports B, C and D
- Ports labelled as B, C and D



I/O Registers

- Input / Output is controlled through special variables called “**registers**”
- Registers are actual hardware memory locations inside the μ Cs with predefined names and sizes
- Assigning a value to these registers in the program changes the corresponding hardware configuration. And, these values can be altered multiple number of time at any point in the program.
- There are 3 registers that control the I/O pins: **DDR, PORT and PIN.**
- Each port has it's own registers. Hence, **DDRC, PORTC, PINC** registers for port C; **DDRB, PORTB, PINB** for port B and likewise

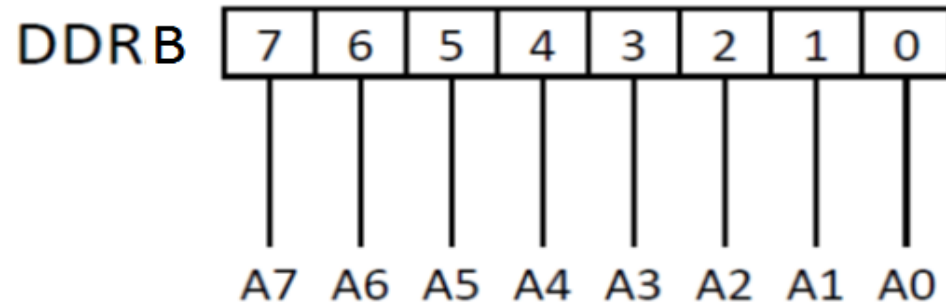
DDR(Data Direction Register)

- Decides whether the pin is Input or Output
- DDR is an 8 bit register. Each bit corresponds to a particular pin on the associated port
- If a bit on the DDR register is **0**, then the corresponding pin on the associated port is set as input
- Similarly, if the bit is **1**, then the pin is set as output
- If a pin is configured as input, then it has some floating voltage unless an external voltage is applied
- For an output pin, the voltage is fixed to a particular value

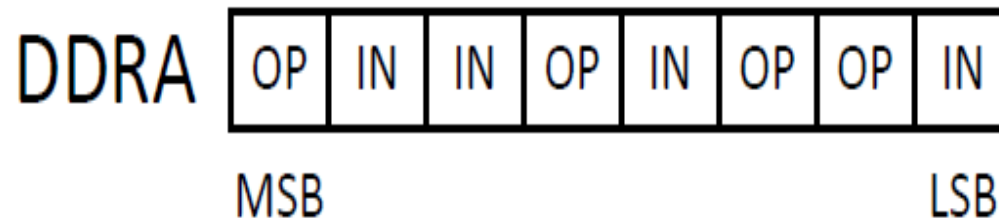


Setting Register Values

- MSB of DDRB corresponds to the pin A7



- If DDRA = 0b10010110, then:



PORT Register

- PORT is also an 8 bit register. The bits on the PORT register correspond to the pins of the associated port in the same manner as in the case of the DDR register.
- PORT is used to set the **output** value.
- If the pin is set as **output**, then a PORT value of 1 will set voltage at that pin to 5V, and PORT value 0 sets the voltage to 0V.
- If the pin is configured as an **input**, PORT value serves the purpose of **pull up** or **pull down**.

PIN Register

- PIN is a register whose value can be read, but cannot be changed inside the program.
- It gives the value of the actual voltage at a particular pin. 1, if the value at the required pin is 5V and 0 for 0V.

Summary

DDR = 0		DDR = 1	
PORT = 0	PORT = 1	PORT = 0	PORT = 1
Pin is input. If unconnected, PIN is 0.	Pin is input. If unconnected, PIN is 1.	Pin is output, value is 0. PIN is always equal to PORT	Pin is output, value is 5V. PIN is always equal to PORT

Some C concepts

- `|` is bitwise OR. Eg. `10100111 | 11000101 = 11100111`
- `&` is bitwise AND. Eg. `10100111 & 11000101 = 10000101`
- `~` is bitwise NOT. Eg. `~10100110 = 01011001`
- `<<` is shift left. `>>` is shift right



Simplest C program for a micro-controller

```
int main(){  
    return 0;  
}
```

Example Program 1

```
#include <avr/io.h>
int main(){
  DDRA = 0b11111111; // or 255 or 0xFF
  while(1){
    PORTA = PINC;
  }
  return 0;
}
```

Example Program 2

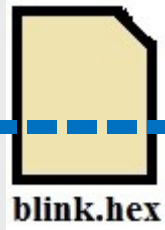
```
#include <avr/io.h>
#include <util/delay.h>
int main(){
  DDRA = 0xFF;
  while(1){
    PORTA = 0xAA;
    _delay_ms(1000);
    PORTA = 0x55;
    _delay_ms(1000);
  }
  return 0;
}
```

How to Program MCU?



CVAVR/AVRSTUDIO

o-> **HOW ?**



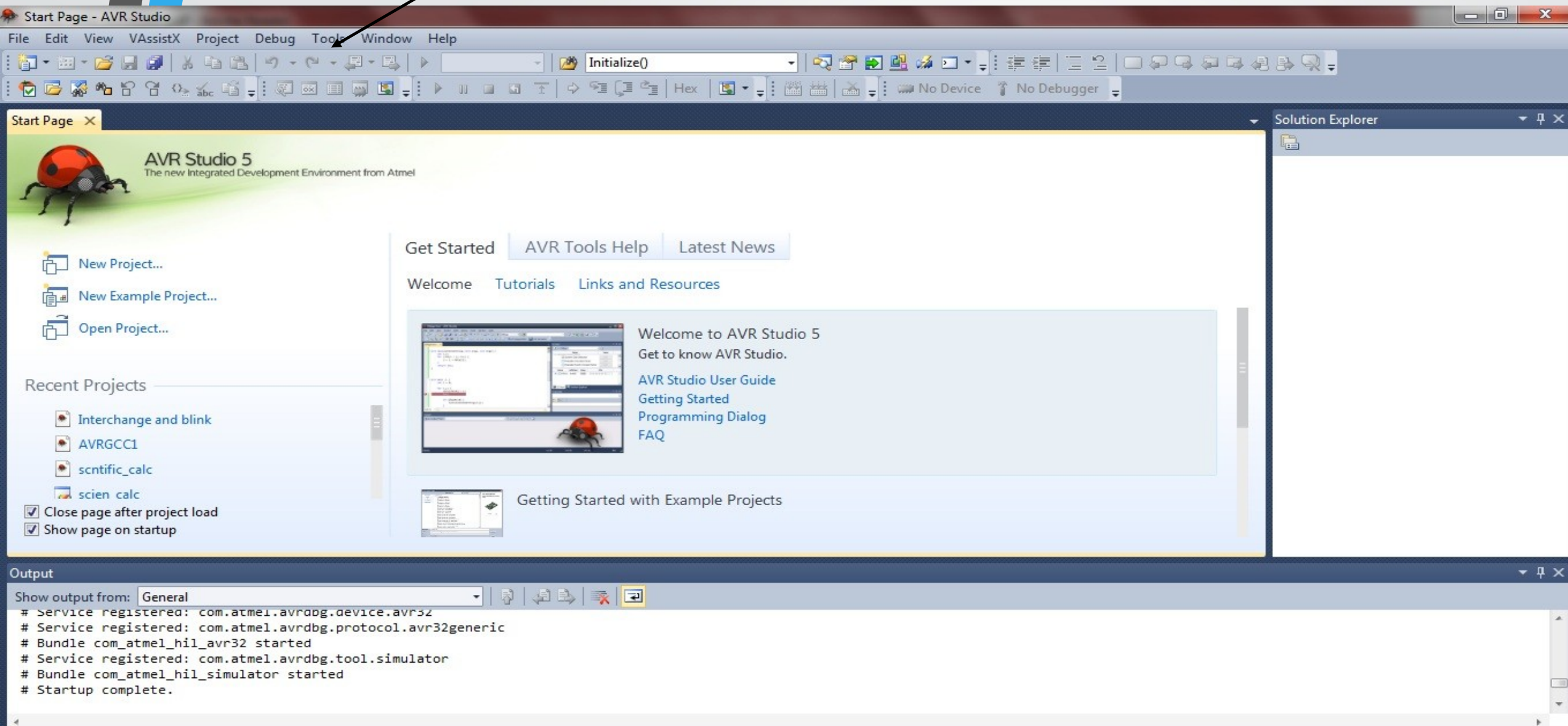
AVRSTUDIO

- #Problem: What kind of files MCU can execute ?
- #Problem: How to transfer that file to MCU ?

AVR Studio



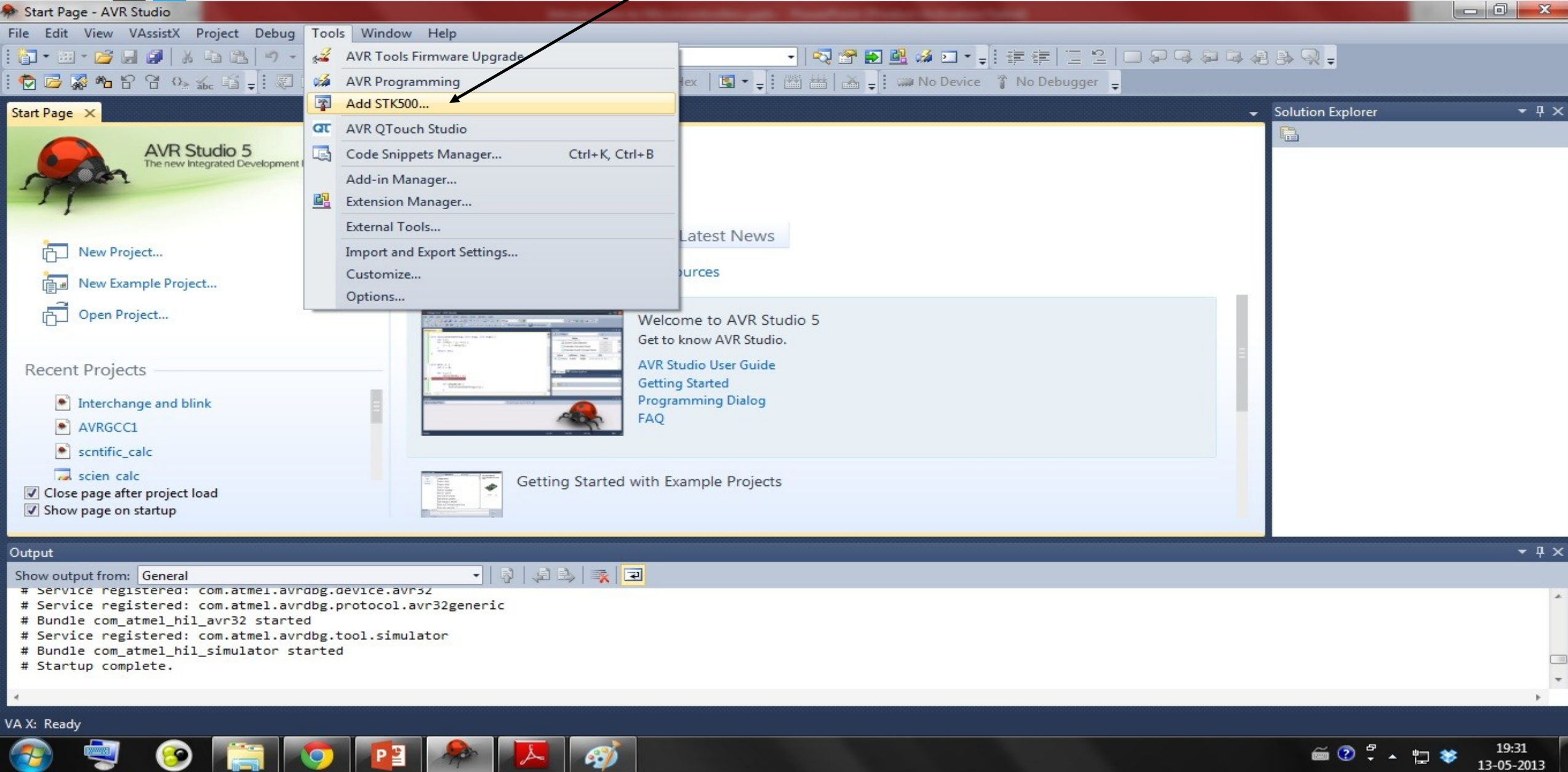
Select Tools



The screenshot shows the AVR Studio 5 Start Page. The title bar reads "Start Page - AVR Studio". The menu bar includes "File", "Edit", "View", "VAssistX", "Project", "Debug", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and development actions. The main content area features a red ladybug logo and the text "AVR Studio 5 The new Integrated Development Environment from Atmel". Navigation tabs include "Get Started", "AVR Tools Help", and "Latest News". A sidebar on the left offers "New Project...", "New Example Project...", and "Open Project...". Below this is a "Recent Projects" list with items like "Interchange and blink", "AVRGCC1", "sctific_calc", and "scien_calc". The main content area has a "Welcome" section with links to "AVR Studio User Guide", "Getting Started", "Programming Dialog", and "FAQ". An "Output" window at the bottom shows the following text:

```
Show output from: General
# Service registered: com.atmel.avrdbg.device.avr32
# Service registered: com.atmel.avrdbg.protocol.avr32generic
# Bundle com_atmel_hil_avr32 started
# Service registered: com.atmel.avrdbg.tool.simulator
# Bundle com_atmel_hil_simulator started
# Startup complete.
```

Select Add STK500



The screenshot shows the AVR Studio 5 Start Page. The 'Tools' menu is open, and the 'Add STK500...' option is highlighted. A black arrow points from the text 'Select Add STK500' to this menu item. The Start Page includes sections for 'New Project...', 'Recent Projects', and 'Welcome to AVR Studio 5'. The Output window at the bottom shows startup logs.

Start Page - AVR Studio

File Edit View VAssistX Project Debug Tools Window Help

AVR Tools Firmware Upgrade

AVR Programming

Add STK500...

AVR QTTouch Studio

Code Snippets Manager... Ctrl+K, Ctrl+B

Add-in Manager...

Extension Manager...

External Tools...

Import and Export Settings...

Customize...

Options...

Start Page X

AVR Studio 5
The new Integrated Development Environment

New Project...

New Example Project...

Open Project...

Recent Projects

- Interchange and blink
- AVRGCC1
- scntific_calc
- scien_calc

Close page after project load

Show page on startup

Welcome to AVR Studio 5
Get to know AVR Studio.

[AVR Studio User Guide](#)

[Getting Started](#)

[Programming Dialog](#)

[FAQ](#)

Getting Started with Example Projects

Output

Show output from: General

```
# Service registered: com.atmel.avrdbg.device.avrs2  
# Service registered: com.atmel.avrdbg.protocol.avr32generic  
# Bundle com_atmel_hil_avr32 started  
# Service registered: com.atmel.avrdbg.tool.simulator  
# Bundle com_atmel_hil_simulator started  
# Startup complete.
```

VA X: Ready

19:31
13-05-2013

Select AVR programming

The screenshot shows the AVR Studio 5 Start Page. The 'Tools' menu is open, and 'AVR Programming' is highlighted. The Start Page includes sections for 'New Project...', 'Recent Projects', and 'Welcome to AVR Studio 5'. The Output window at the bottom shows startup logs.

Start Page - AVR Studio

File Edit View VAssistX Project Debug Tools Window Help

- AVR Tools Firmware Upgrade
- AVR Programming**
- Add STK500...
- AVR QTouch Studio
- Code Snippets Manager... Ctrl+K, Ctrl+B
- Add-in Manager...
- Extension Manager...
- External Tools...
- Import and Export Settings...
- Customize...
- Options...

Start Page x

AVR Studio 5
The new Integrated Development Environment

New Project...
New Example Project...
Open Project...

Recent Projects

- Interchange and blink
- AVRGCC1
- scntific_calc
- scien_calc

Close page after project load
 Show page on startup

Welcome to AVR Studio 5
Get to know AVR Studio.
AVR Studio User Guide
Getting Started
Programming Dialog
FAQ

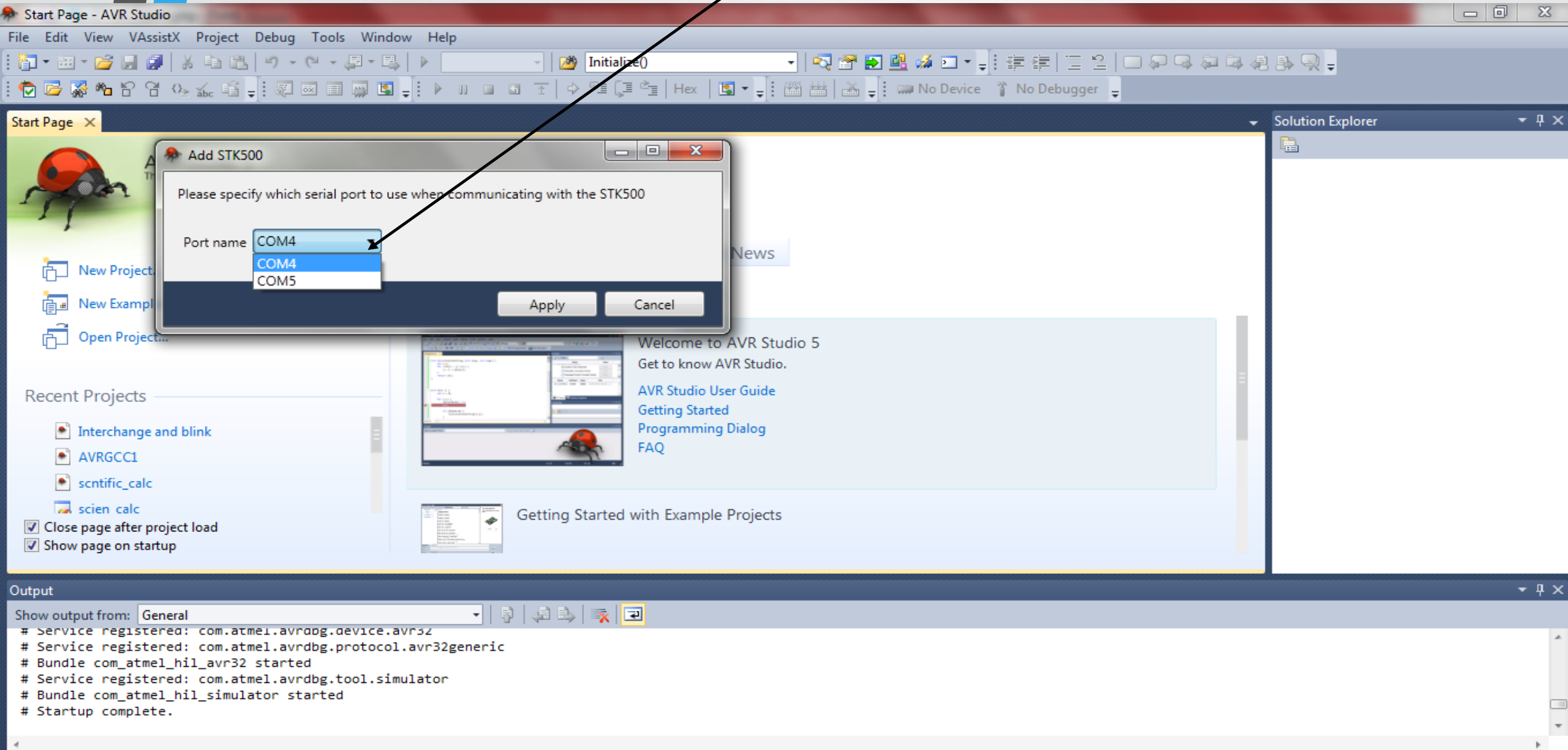
Getting Started with Example Projects

Output

Show output from: General

```
# Service registered: com.atmel.avrdbg.device.avrsz  
# Service registered: com.atmel.avrdbg.protocol.avr32generic  
# Bundle com_atmel_hil_avr32 started  
# Service registered: com.atmel.avrdbg.tool.simulator  
# Bundle com_atmel_hil_simulator started  
# Startup complete.
```

Select COM port



The screenshot displays the AVR Studio 5 interface. A dialog box titled "Add STK500" is open, prompting the user to "Please specify which serial port to use when communicating with the STK500". The "Port name" dropdown menu is open, showing "COM4" and "COM5" as options. An arrow points from the text "Select COM port" to the dropdown menu. The background shows the AVR Studio 5 Start Page with various options like "New Project", "New Example", and "Open Project...". The Output window at the bottom shows the following text:

```
Show output from: General
# Service registered: com.atmel.avrdbg.device.avr32
# Service registered: com.atmel.avrdbg.protocol.avr32generic
# Bundle com_atmel_hil_avr32 started
# Service registered: com.atmel.avrdbg.tool.simulator
# Bundle com_atmel_hil_simulator started
# Startup complete.
```

Select Device -> Click Apply -> Read Device ID -> Read target Voltage -> Choose Hex File -> Then Program

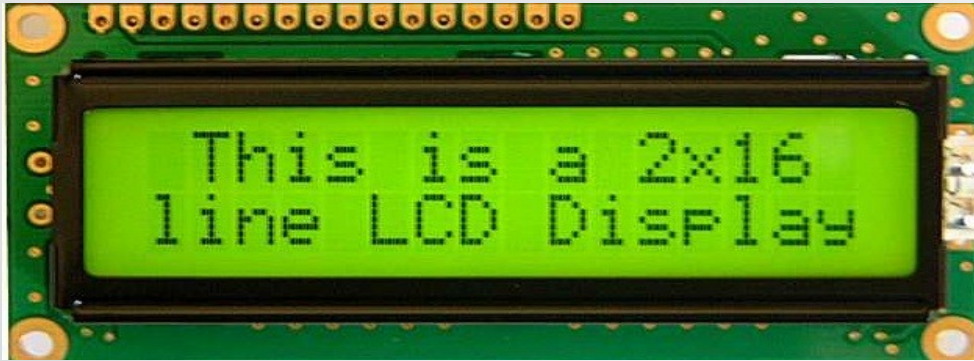
The screenshot shows the AVR Studio 5 IDE with the AVR Programming dialog box open. The dialog box has several sections and controls:

- Tool:** STK500
- Device:** ATmega32
- Interface:** ISP
- Device ID:** not read
- Target Voltage:** ---

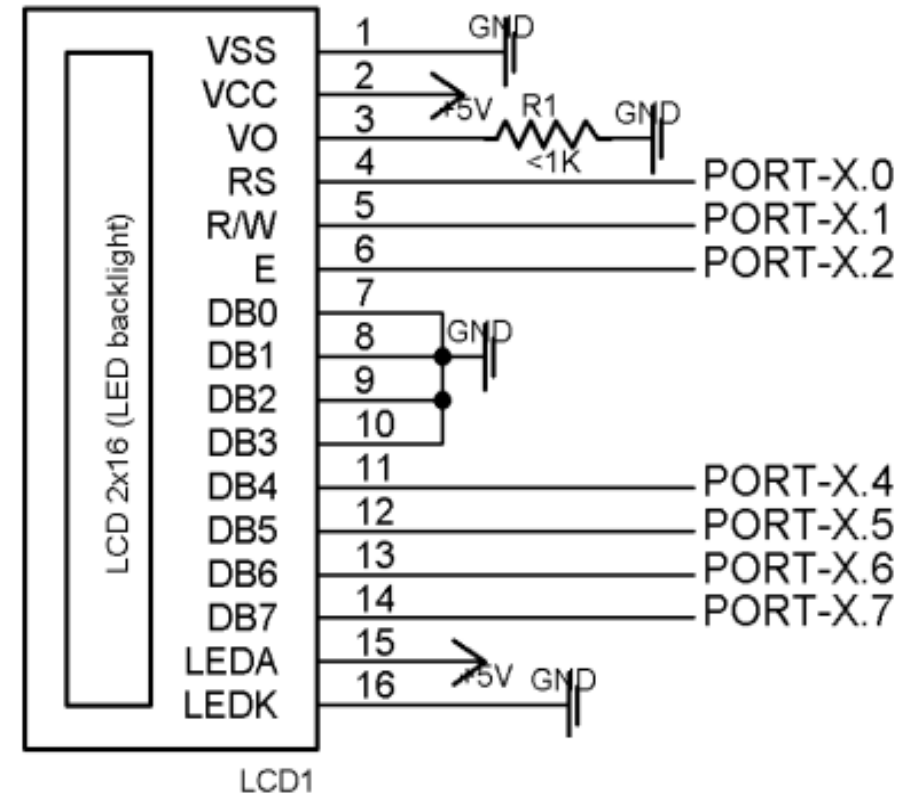
Buttons in the dialog include **Apply**, **Read**, **Erase Device**, **Program**, **Verify**, and **Read...**. The **Flash** section shows a file path: `C:\Users\SHIVENDU's\scntific_calc\default\scntific_calc.hex`. The **EEPROM** section is empty.

The **Output** window at the bottom left shows error messages:

```
Getting clock value...Failed!  
# Failed to open COM3. Error 0x2.  
19:49:48.452: [ERROR] TCF command: Tool:com  
# Failed to open COM3. Error 0x2.  
19:49:48.800: [ERROR] TCF command: Tool:com  
19:49:48.801: [ERROR] Unable to connect to  
# Failed to open COM3. Error 0x2.
```

- We interface an LCD to our microcontroller so that we can display messages, outputs, etc.
- Sometimes using an LCD becomes almost inevitable for debugging and calibrating the sensors
- We will use the 16x2 LCD, which means it has two rows of 16 characters each. Hence in total we can display 32 characters



IR - TSOP Pair!



Just Think Over!

- TSOP sensor detects the presence of light from the Infrared LED
- How will it distinguish from other Infrared light already present
- Should we use some kind of encoding ?
- TSOP sensor detects Infrared light only at 38 KHz
- How do we generate light at 38KHz?
- Timers and Interrupts... we will talk in the next lecture

Thanks ㄴ



Shivendu Bhushan
351 / Hall 2
shivendu@iitk.ac.in
8960419775



Sonu Agarwal
368 / Hall 3
sonuagr@iitk.ac.in
9005888958



Swapnil Upadhyay
F-102 / Hall 5
swapnilu@iitk.ac.in
7417279731, 7379599283

Website : <http://students.iitk.ac.in/eclub/index.php>

FB Group : <https://www.facebook.com/groups/eclub.iitk/>

E-mail : eclub.iitk@gmail.com

Youtube : <http://www.youtube.com/user/electronicsclub>