

Summer Camp '13



Introduction to Microcontrollers

Shivendu Bhushan

Embedded Systems

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

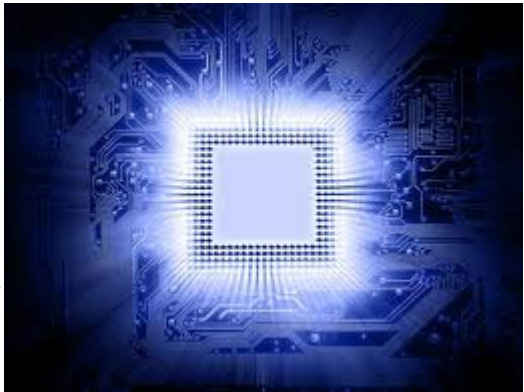
Example: Music player



Output



Input



Controller



Output

Storage Device

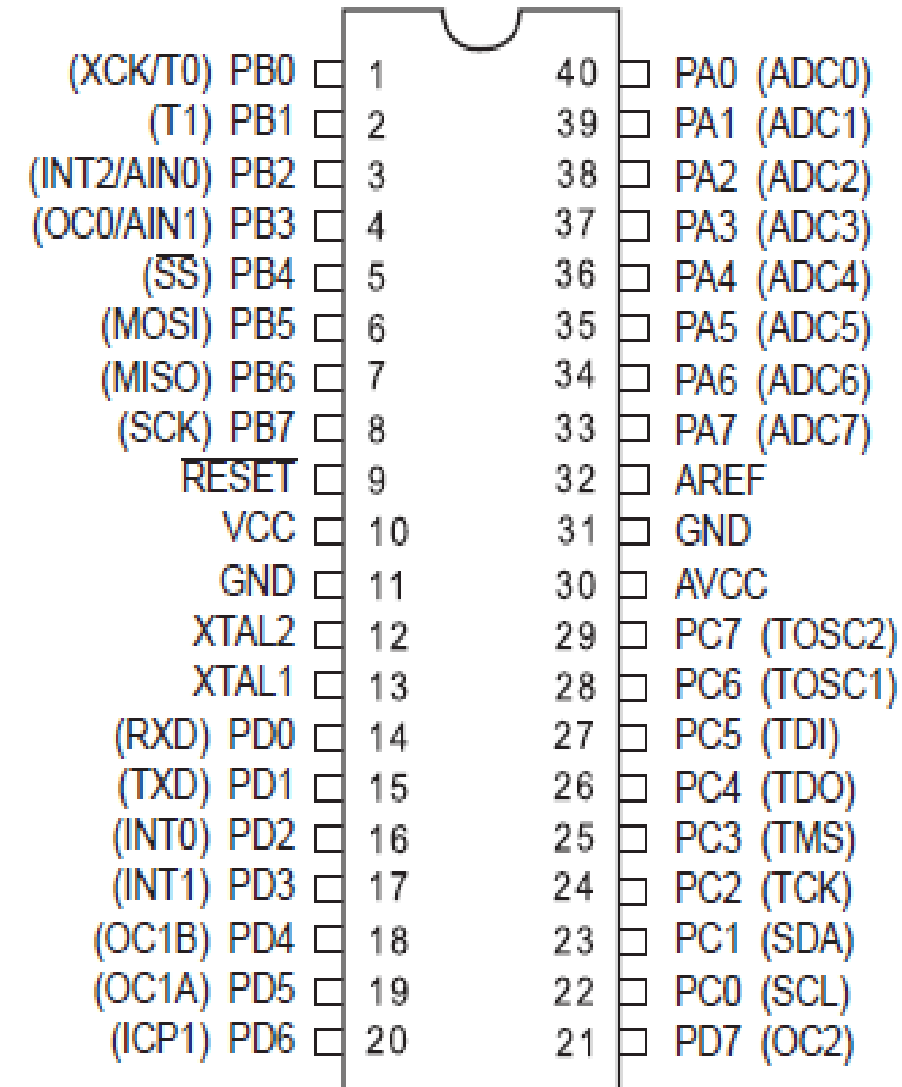


The Microcontrollers

- 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.

Example: Atmega 16

- 40 pin IC.
- 32 pins for I/O.
- 8 pins reserved.
- I/O pins divided into 4 groups of 8 pins, called ports.
- Ports labeled as A, 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, **DDRA, PORTA, PINA** registers for port A; **DDRB, PORTB, PINB** for port B and likewise.

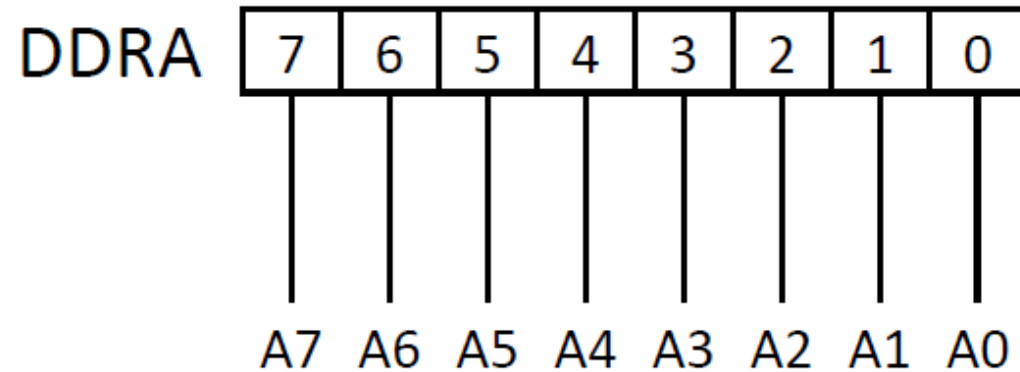
Registers

DDR(Data Direction Register)

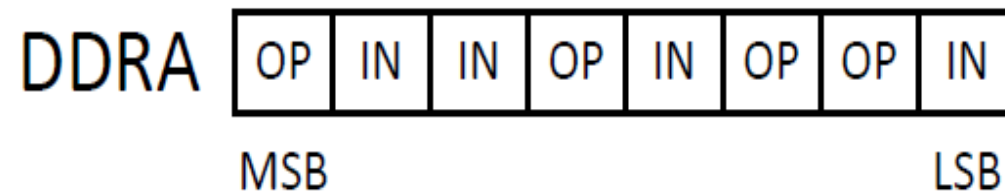
- Decides whether the pin is Input or Output.
- DDR is an 8 bit register. Each bit corresponds to a particular pin on the associated port.
- If a bit on the DDR register is 0, then the corresponding pin on the associated port is set as input.
- Similarly, if the bit is 1, then the pin is set as output.
- 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 DDRA 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.

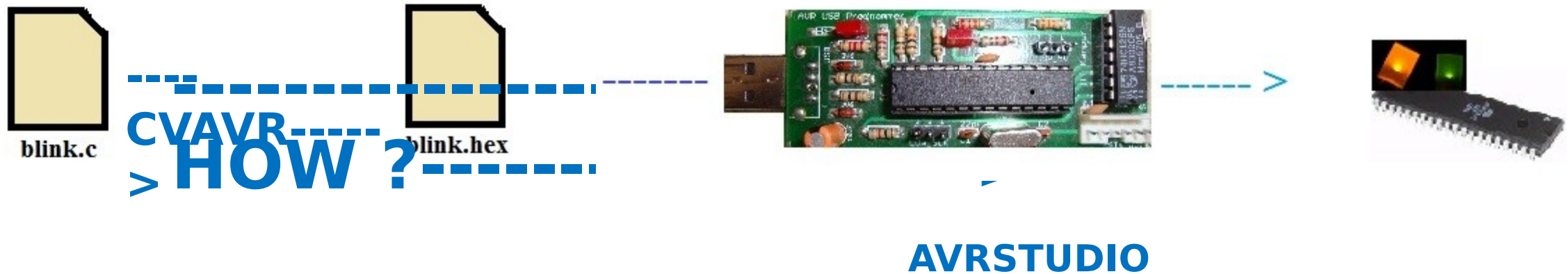
Summary

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

Applications(Takneek/Techkriti)

- Line Following Robots
- Wireless keyboard
- Wireless Gamepad
- Tachometer
- Music synthesizer
- Motion sensing games

How to program MCU ?



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

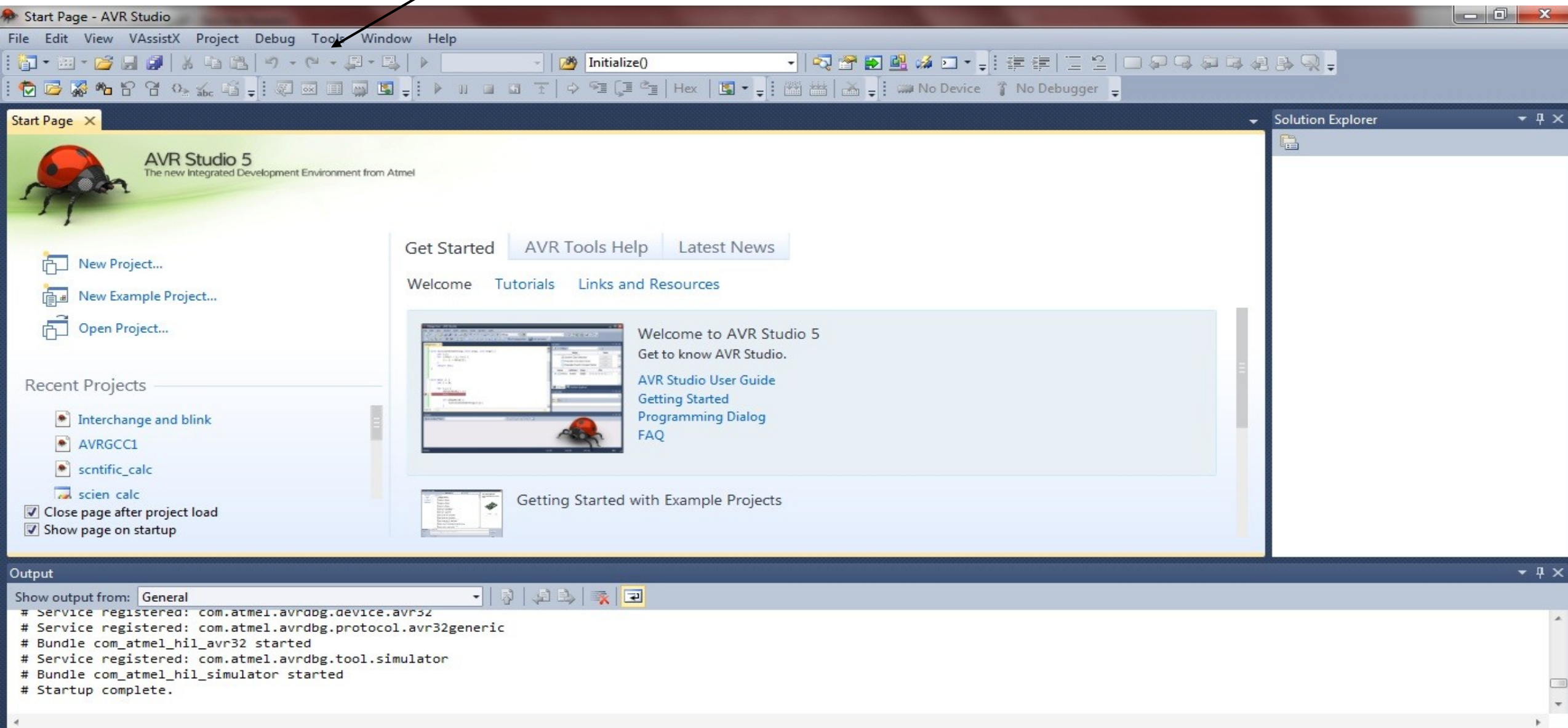
Software needed:

- **CVAVR**----> Editor and compiler
- **AVR Studio**---- > Transfer the code to Atmega
- **ISIS**--- > Circuit Simulator

AVR Studio



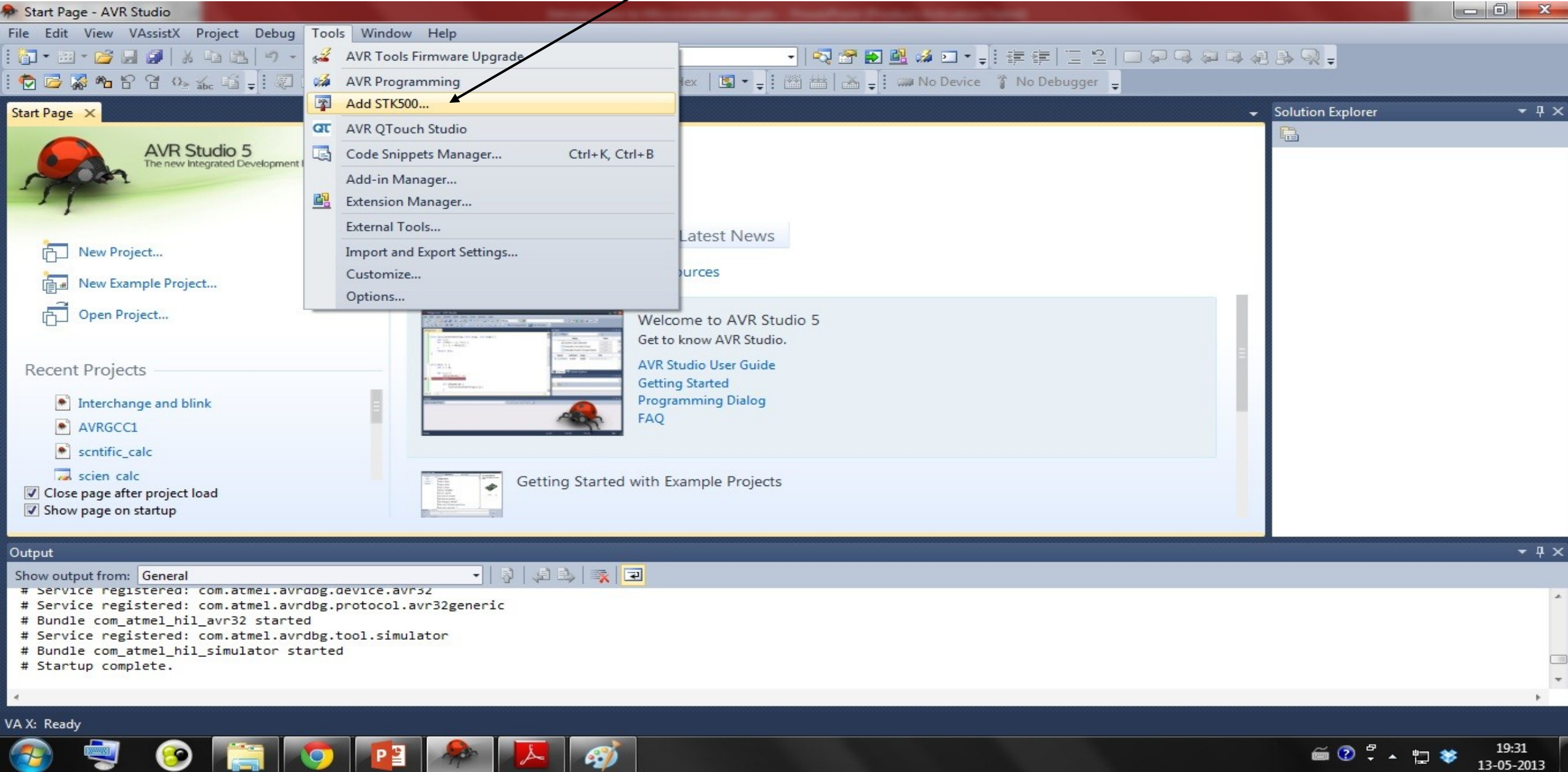
Select Tools



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..." options, along with a list of "Recent Projects" such as "Interchange and blink", "AVRGCC1", "sctific_calc", and "scien_calc". The main content area displays a "Welcome to AVR Studio 5" message with links to "AVR Studio User Guide", "Getting Started", "Programming Dialog", and "FAQ". Below this is a section titled "Getting Started with Example Projects". The bottom of the window shows an "Output" window with 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' at the top of the image 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 system messages.

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

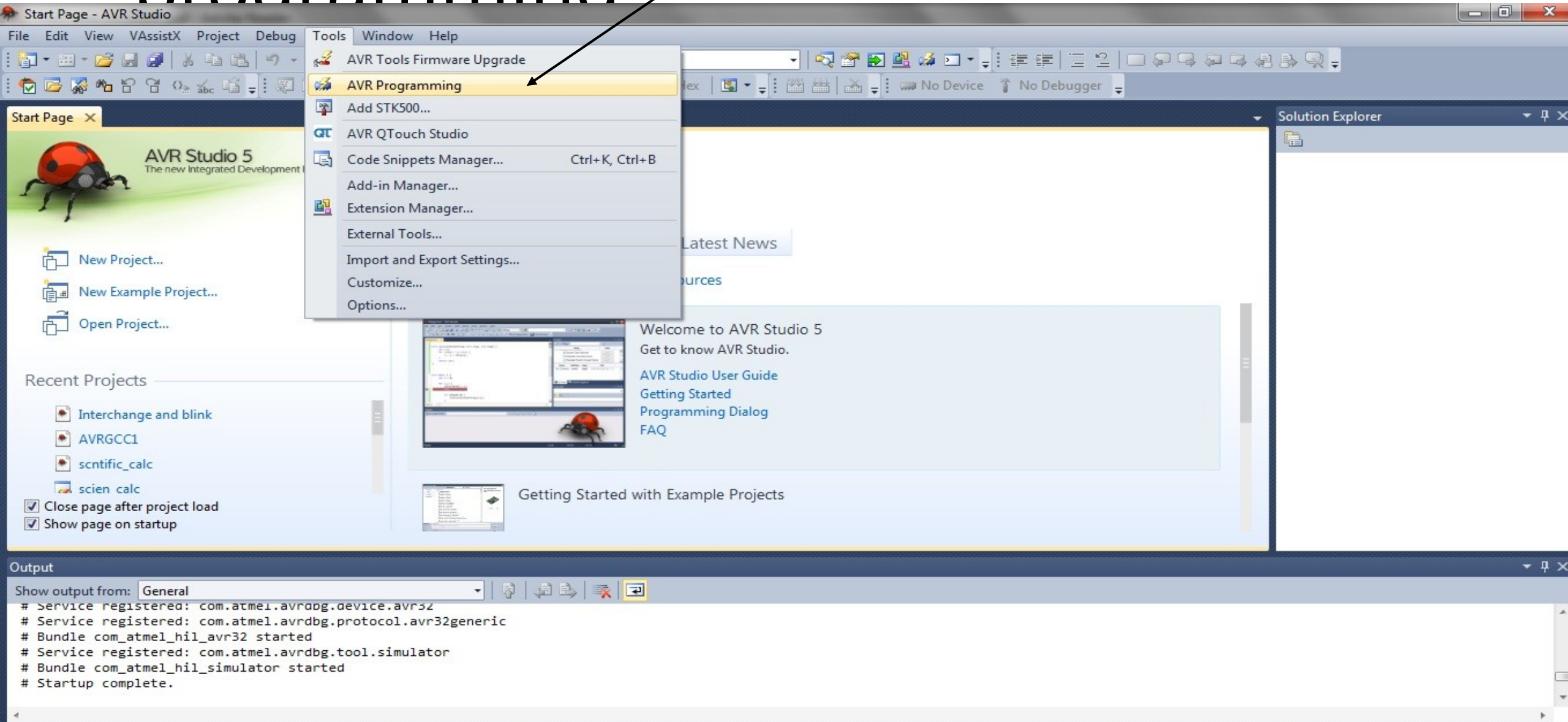
```
# Service registered: com.atmel.avrdbg.device.avr32  
# Service registered: com.atmel.avrdbg.protocol.avr32generic  
# Bundle com_atmel_hil_avr32 started  
# Service registered: com.atmel.avrdbg.tool.simulator  
# Bundle com_atmel_hil_simulator started  
# Startup complete.
```

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. An arrow points from the word 'programming' in the title to this menu item. The Start Page includes sections for 'New Project...', 'Recent Projects', and 'Getting Started with Example Projects'. The Output window at the bottom shows the following text:

```
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" selected, with "COM4" and "COM5" as visible options. An arrow from the text "Select COM port" points to the dropdown menu. The background shows the "Start Page" with 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 shows error messages: "Getting clock value...Failed!" and "Unable to connect to".

Arrows from the text above point to the following elements in the dialog:

- Select Device:** Points to the **Device** dropdown menu.
- Click Apply:** Points to the **Apply** button.
- Read Device ID:** Points to the **Read** button.
- Read target Voltage:** Points to the **Target Voltage** dropdown menu.
- Choose Hex File:** Points to the file selection button (three dots) next to the **Flash** field.
- Then Program:** Points to the **Program** button.

Thanks...!!