



Micro 2 Lab Assignments

Rules and Helpful tips

TA: Ioannis Smanis

Overview

- 1. TA Info**
- 2. Lab Assignments General Info**
- 3. Lab Report Rules**
- 4. Lab Rules & Warnings !!**
- 5. Recommended Materials**
- 6. Tools & Tips: USB-to-UART cable**
- 7. PIC Project requirements**
- 8. Working with a PIC**
- 9. Intel Galileo Projects**
- 10. Working with Intel Galileo Gen 2**

TA Info

TA:	Ioannis Smanis
TA Office hours :	Monday, Tuesday, Wednesday - 2.00pm to 5.00pm
TA Office Location:	Ball Hall, Room #402, Cubicle #48
TA Mailbox:	Ball Hall, Room #301
TA e-mail:	<u>ioannis_smanis@student.uml.edu</u>
Deliverables (ONLY) e-mail:	<u>micro2tamailbox@gmail.com</u>

- Please do not hesitate to discuss with me any difficulty that you cope with
- See me at my office during my office hours or e-mail me for anything related with your Lab assignments and I will find a solution for you as fast as I can !
- Please, expect quick e-mail responses **ONLY** during my office hours.
- Please , **ALWAYS** refer your Group # in the subject of your e-mail (i.e.: subject: “**Group #05 clarification on lab X**”)
- Please do not hesitate to bother me out of my office hours **ONLY WHEN** you have an **urgent issue** with your lab assignment (defective parts, need more components, fried parts, etc.). In this case, e-mail me with subject: “**URGENT - Group #xx**” or visit me at my office space
- Please, **NO QUESTIONS** about mid-term exam and final exam are allowed.

Lab Assignments General Info

- Any announcement, update or Lab assignment adjustment is going to be posted on “**Piazza**” (<https://piazza.com/class/j5fnal0lsvb5af>) *Make sure you get Piazza notification e-mails*
- **You are responsible to follow Piazza** discussion threads or announcement threads
- Lab assignments materials will be uploaded on **GitHub** repository: <http://github.com/yanluo-uml/micro2.git>
- Lab assignments are described in the **pdf documents**
- **New lab assignment** will usually be released few days before your prior lab demo time
- **It is your responsibility** to review all given materials from GitHub repo
- **It is your responsibility** to choose the right partners for your team

Lab Reports Rules

- **Lab Report Materials:**

1. Hard-copy: **a printed copy of the PDF document** named “**LabX_name_lastname.pdf**” (*Lab Report document format will be specified on Piazza*)
2. Soft-copy: a **ZIP** file with name “**LabX_GroupXX.zip**”
 - + All group-members PDF documents (all team members actual PDF documents)
 - + Shared files exported in *a common readable/acceptable file format* (jpg,png, pdf) which you utilized to complete your lab project: *circuit design CAD files (fritzing, Eagle CAD, etc), schematics, images, source code files (in .c/.h file format or the entire project file structure as it was created)*

Deliverables: *both Hard-copies and Soft-copies are required to be delivered on time!!!*

- **Hard-copies :** should be delivered in TA’s physical mailbox **on time** according to the **specified project deadline**.
- **Soft-copies :** should be sent via e-mail to **micro2tamailbox@gmail.com** **on time** according to the **specified project deadline**.

e-mail format:

Subject: *Lab X Group XX*

e-mail content: *LabX_GroupXX.zip (just the zip file)*

Lab Reports Rules

- **Lab Report Materials:** Non-complete Lab report soft-copies will be rejected automatically.
- **Lab Report Deadlines:**

Soft-copies and Hard-copies should be delivered by each **Monday 10am**, 4 days after the last demo day. For important dates review Micro 2 course syllabus on Piazza
- **Delayed Lab Report Materials** will be penalized with 25% of the total earned points

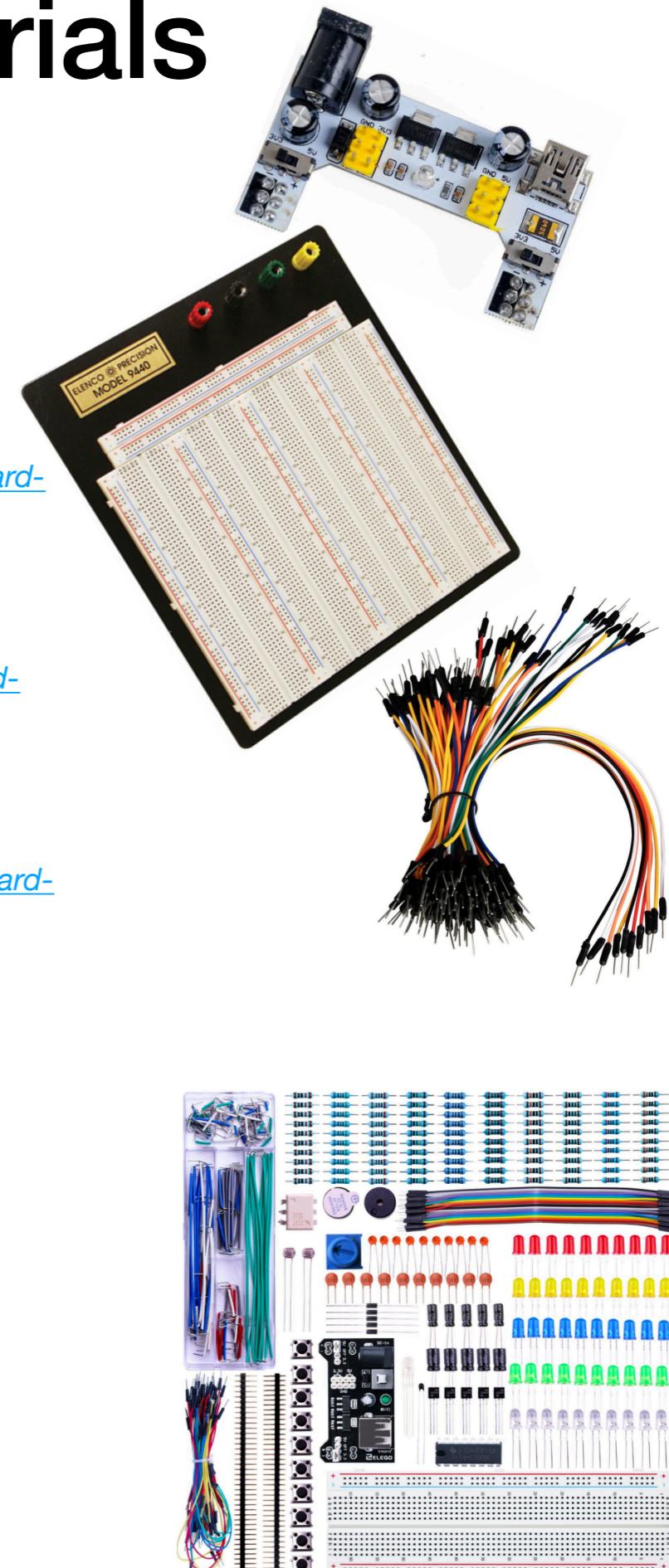


Lab Rules & Warnings !!

- Students are **responsible** for any damage to the given Lab Hardware kit
- Students **should return** the Lab Hardware Kit with fully functional components
- If you by mistake **fry or brick** any component, you should order replacement parts before you return your Lab Hardware Kit back
- A Lab Report is **an individual job** - each student should write his/her own lab report document
- **Shared Files between group-members** are **ONLY:** CAD file, schematics, pictures, voltage/signal readings measurement tables, source code
- **Copied-Pasted** content in the lab reports will be **penalized** from **other students** or from **provided documents**

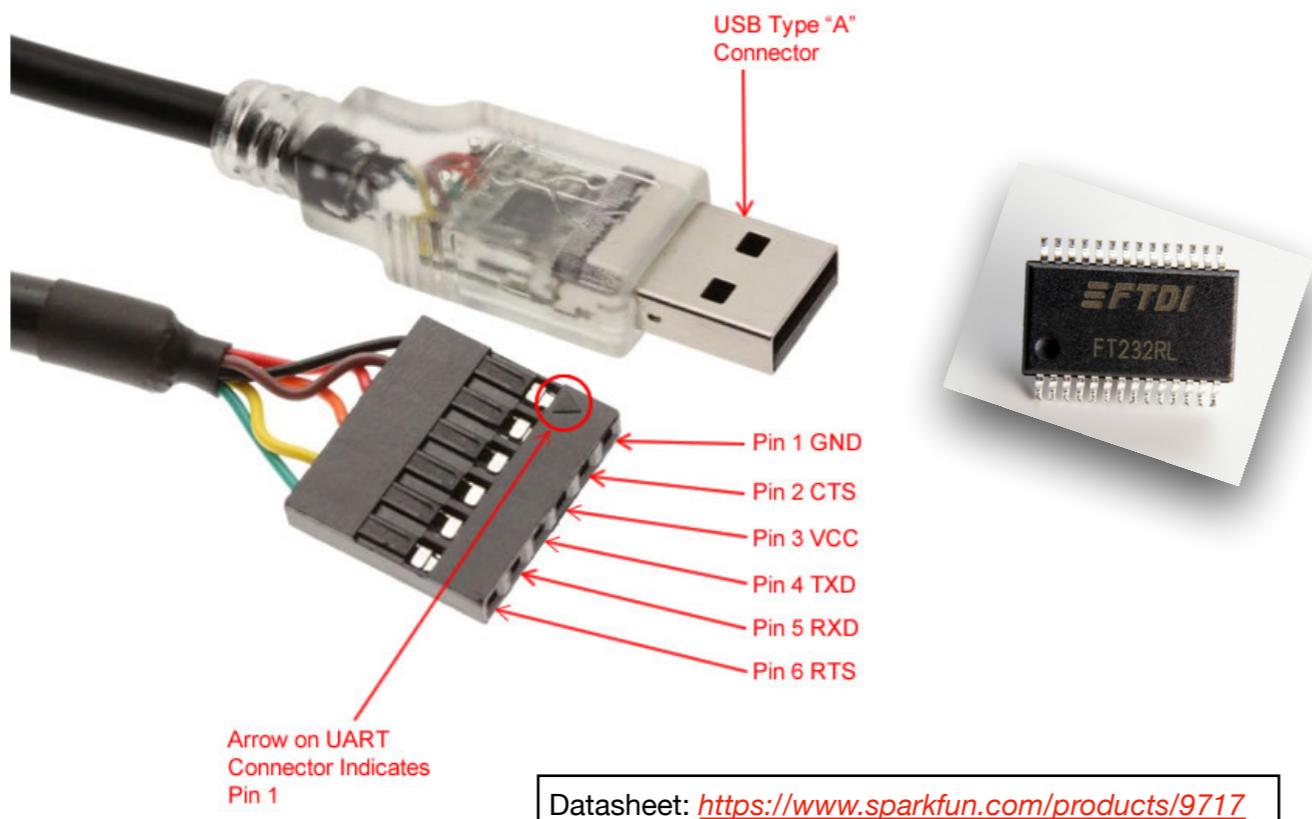
Recommended Materials

- **Breadboard** source: https://www.amazon.com/Wangdd22-Solderless-Breadboard-Protoboard-Tie-point/dp/B01N41WW61/ref=sr_1_3?ie=UTF8&qid=1505235805&sr=1-3&keywords=solderless+Breadboard+Protoboard
- **Solid Core wires** source: https://www.amazon.com/Makerfocus-140pcs-Breadboard-Board-Jumper/dp/B01M36HQWN/ref=sr_1_16?ie=UTF8&qid=1505235298&sr=1-16&keywords=breadboard+wires
- **Jumper cables** source: https://www.amazon.com/Haitronic-Multicolored-Breadboard-Arduino-raspberry/dp/B01LZF1ZSZ/ref=sr_1_3?ie=UTF8&qid=1505235183&sr=1-3&keywords=jumper+cables
- **3.3V Power Supply** source: https://www.amazon.com/WINGONEER-Breadboard-Supply-Arduino-Solderless/dp/B06XXXGFJ1/ref=sr_1_22?ie=UTF8&qid=1505235992&sr=1-22&keywords=breadboard+3.3v+supply
- **Recommended Starter Kits:**
[\(Recommended\) https://www.amazon.com/16Hertz-Electronics-Breadboard-Resistors-Raspberry/dp/B00J4RN61A/ref=sr_1_7?ie=UTF8&qid=1505234890&sr=8-7&keywords=electronics+starter+kit](https://www.amazon.com/16Hertz-Electronics-Breadboard-Resistors-Raspberry/dp/B00J4RN61A/ref=sr_1_7?ie=UTF8&qid=1505234890&sr=8-7&keywords=electronics+starter+kit)
[\(Recommended\) https://www.amazon.com/Solderless-Breadboard-jumper-supply-connector/dp/B01M11AVG8/ref=sr_1_21?ie=UTF8&qid=1505235045&sr=8-21&keywords=breadboard](https://www.amazon.com/Solderless-Breadboard-jumper-supply-connector/dp/B01M11AVG8/ref=sr_1_21?ie=UTF8&qid=1505235045&sr=8-21&keywords=breadboard)



Tools & Tips: USB-to-UART cable

- What is this ?



What software you will need ?

MacOSX/Windows/Linux systems:
FTDI drivers

&

Putty.exe for Windows
Terminal app for Mac OSX/Linux

- An a USB RS-323 Serial port adapter at 3.3V



A screenshot of a terminal window titled "ch208a.cae.tntech.edu - PuTTY". The window displays a Linux login session. The user "mwr" logs in using keyboard-interaction authentication. The terminal shows the following text:

```
login as: mwr
Using keyboard-interactive authentication.
Password:
Linux ch208a 2.6.8-2-686-smp #1 SMP Tue Aug 16 12:08:30 UTC 2005 i686 GNU/Linux

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

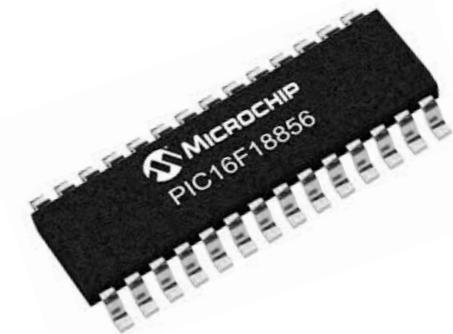
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
No mail.

Last login: Mon May  1 13:49:31 2006 from ch314c.cae.tntech.edu
mwr@ch208a:~$
```

PIC Project requirements

Project (LAB 1): LDR sensor + Servo Motor + LED + PIC micro-controller

Easy Project Debugging: use *UART to Serial cable (Console output)*



Working with a PIC micro-controller:

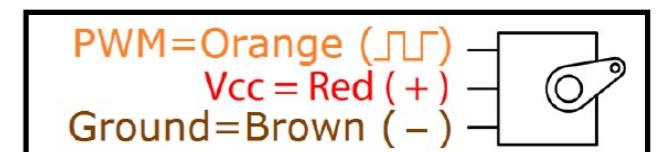
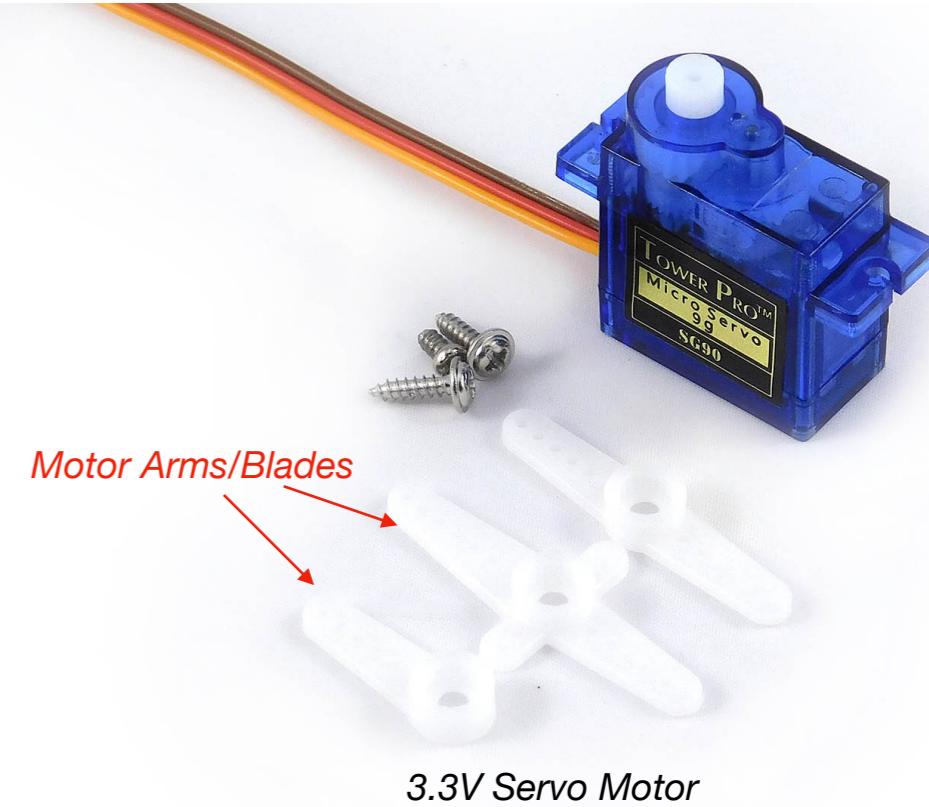
- You need a **PIC Kit 3** Debugger/Programmer (*PICKit3 wiring is provided via specified document*), an 8bit **PIC** micro-controller and MPLAB X development environment



Programming a PIC microcontroller

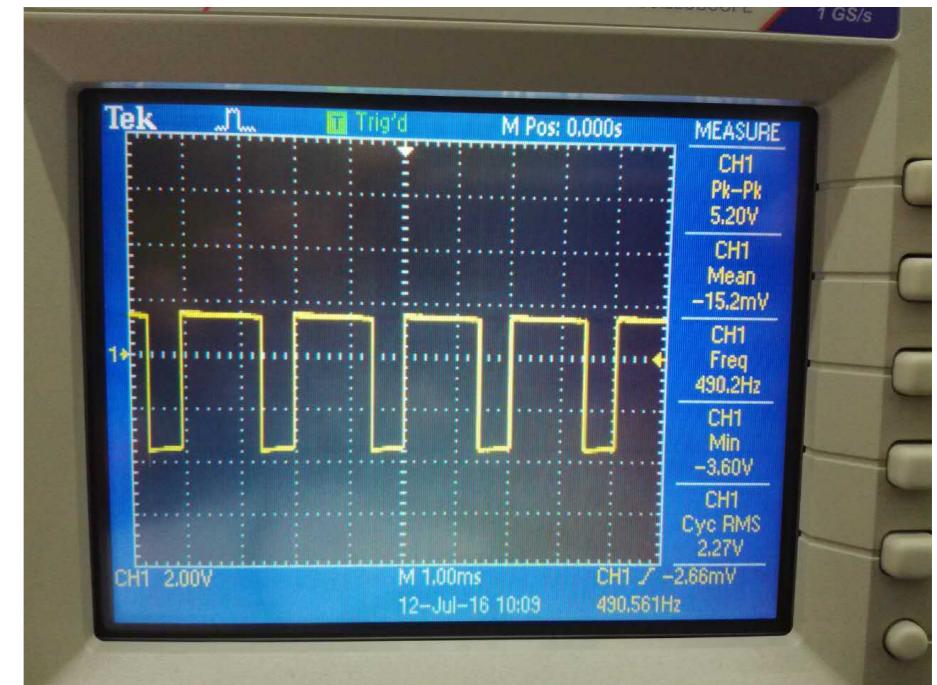
PIC Project requirements

- Servo Motor: **PWM** Signal Input
- Keep **PWM** output **frequency** low (*close to 50Hz*) using wait/delay functions
- Change the **Duty Cycle** to set the motor arm to a specific position
- Servo motors **does not** spin around. Rotation range is up to 180 degrees (90 degrees for each direction +/-)
- Refer to its datasheet on git repository



PIC Project - Tips

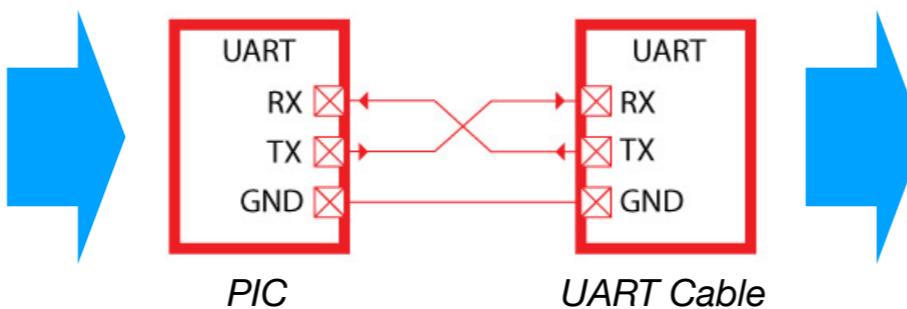
- How to test your PWM output signal from PIC?
- Increase/Decrease Duty Cycle and see if the square wave pulse changes



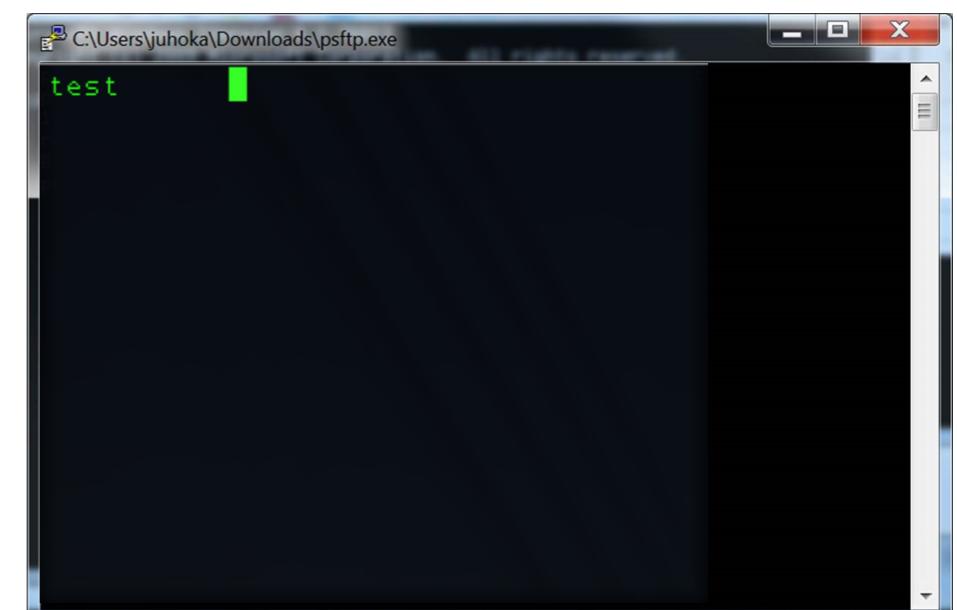
check your PWM signal on the Oscilloscope

- Debugging a PIC application via Serial Cable:

```
void main(void) {  
    SYSTEM_Initialize();  
    printf("Test \n\r");  
}
```



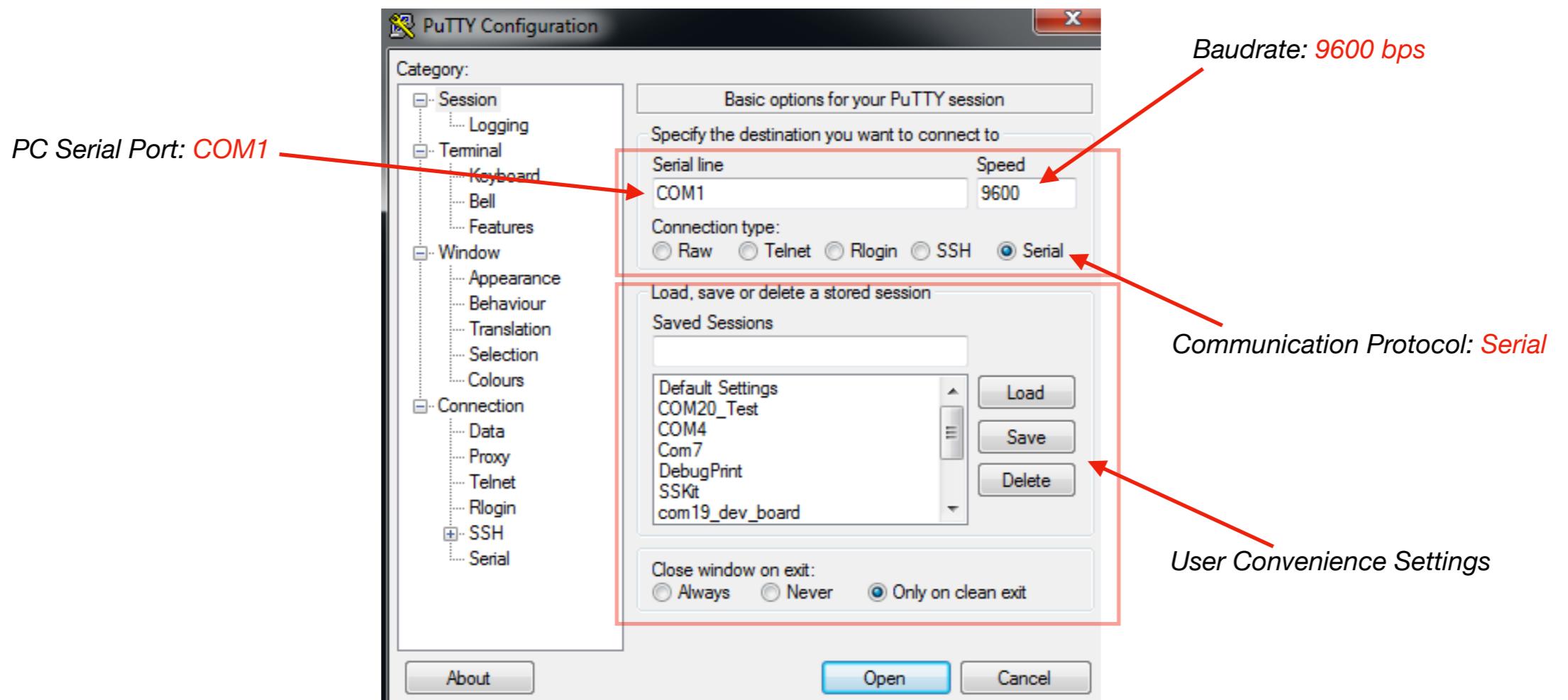
main.c code in MPLAB X IDE:
debugging statement



Putty Client: printed output context

PIC Project - Tips

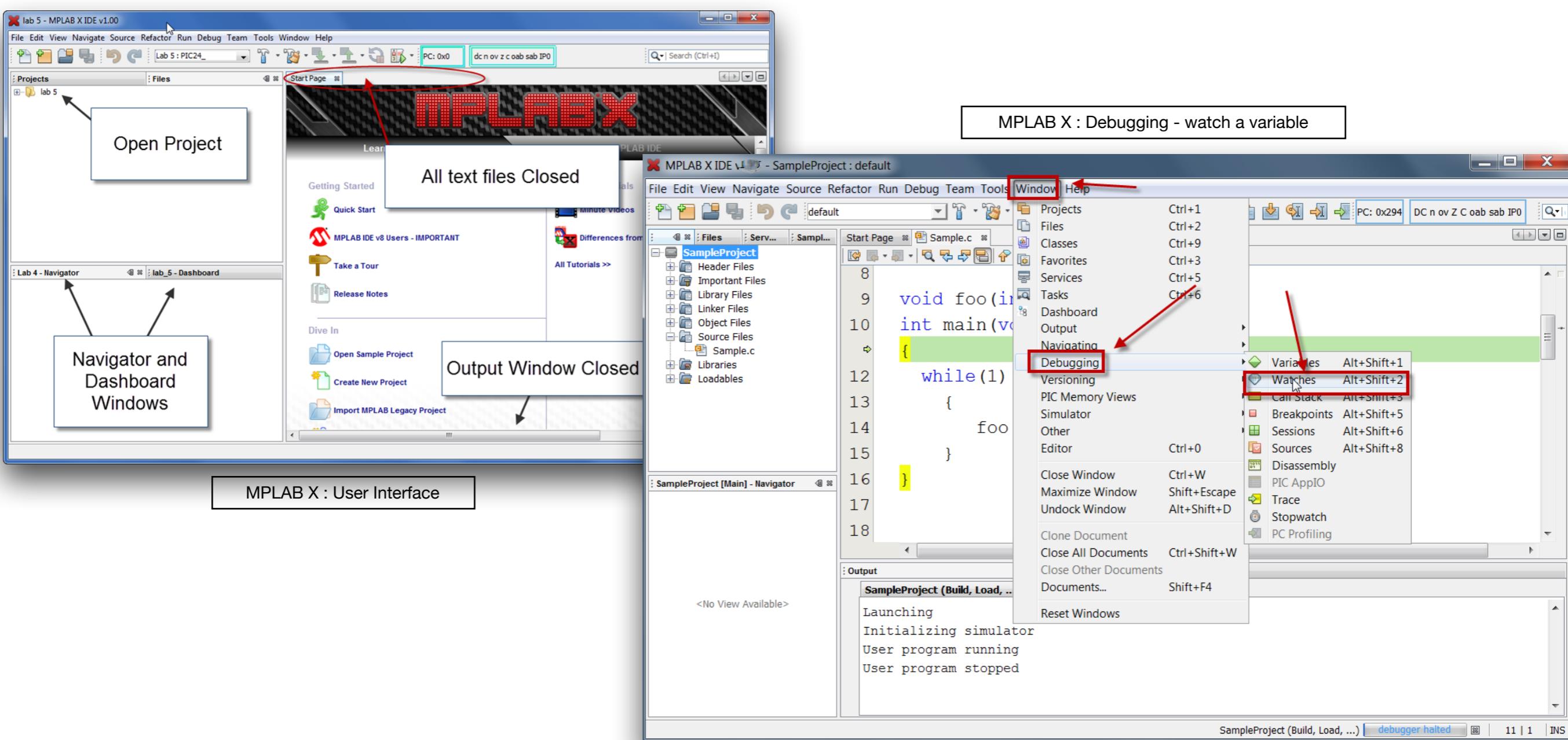
- Setting up a Putty console PC application for Serial port Mode:



Working with a PIC

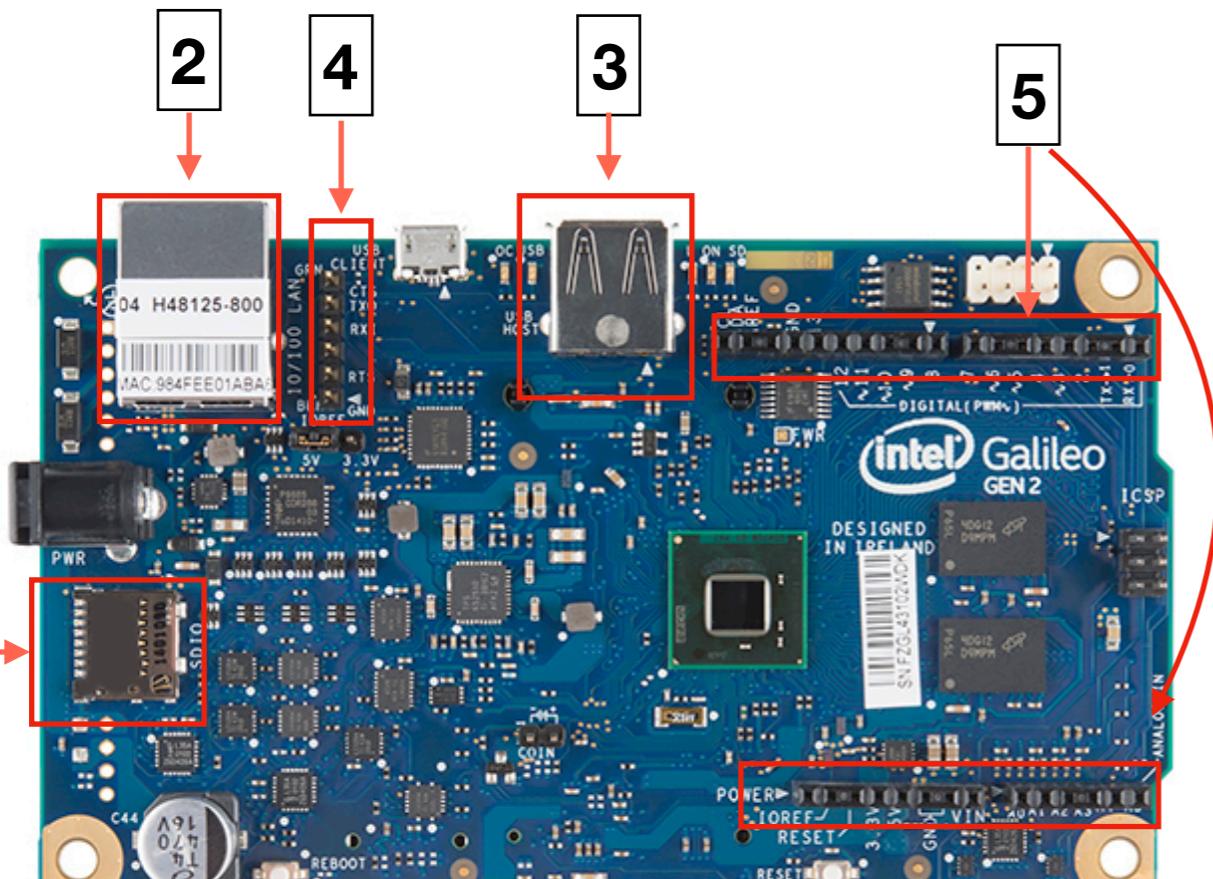
Needed Software: MPLAB X IDE + XC8 compiler

MPLAB X IDE: <http://www.microchip.com/mplab/mplab-x-ide>
XC8 compiler: <http://www.microchip.com/mplab/compilers>



Intel Galileo Projects

- Embedded-linux x86-32bit platform: Yocto OS
- Yocto has **terminal-based** User Interface
- C/C++, Python compilers are already installed
- Package manager is not available



Intel Galileo Board Gen 2

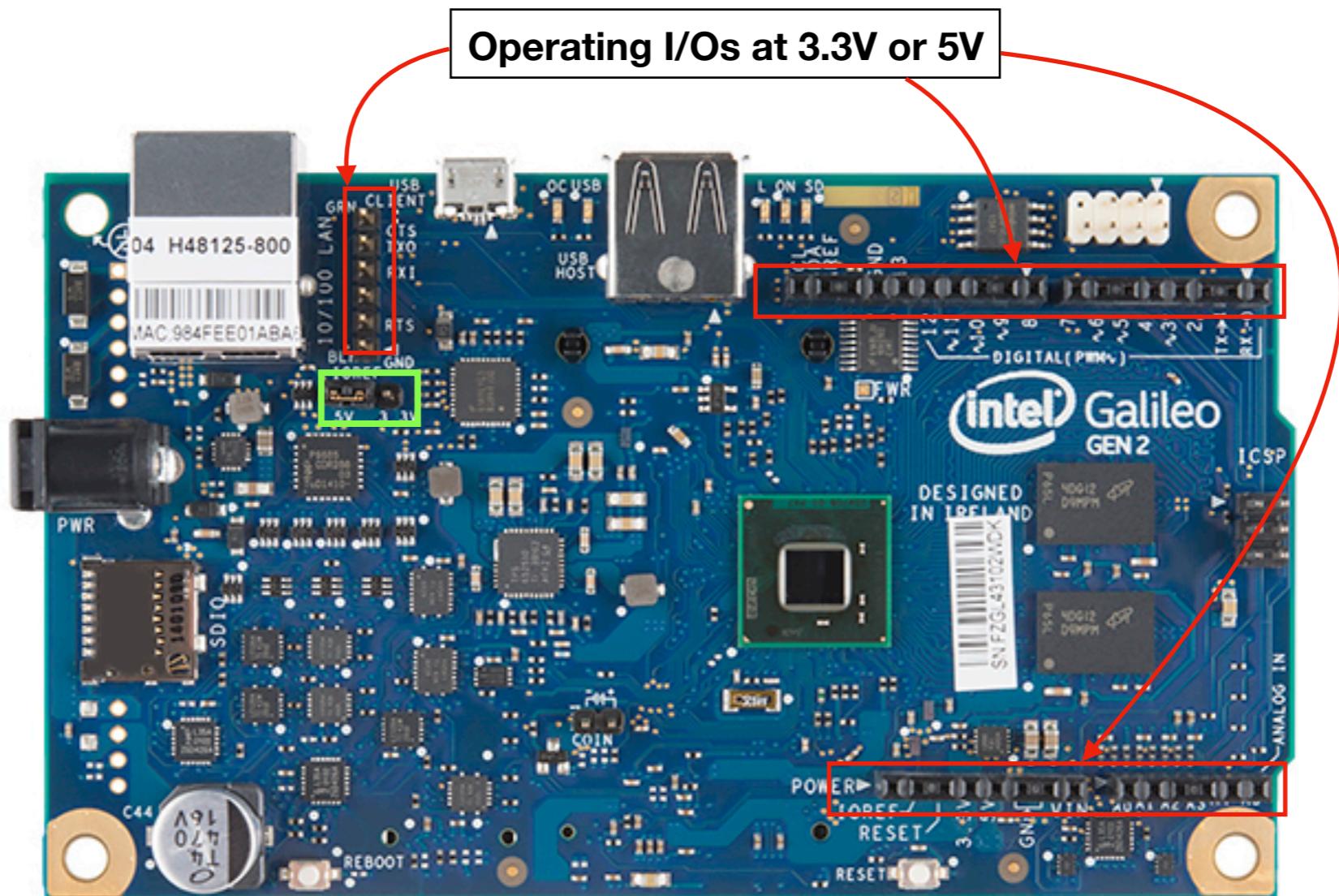
Basic Hardware Features Use:

1. **SD Card:** You should install Yocto OS (***Do not remove SD Card while the board is ON***)
2. **Ethernet Port and Wi-Fi Card installed:** connect the board to the internet and use SSH protocol to exchange files remotely
3. **USB Host Port:** use USB flash drives (**FAT32 formatted**) to transfer files
4. **UART Port:** use the FTDI Cable for console output
5. **GPIO ports:** programmable general purpose I/O pins

Working with Intel Galileo Gen 2

Set up the correct operating voltage:

WARNING: use 3.3V (NOT 5V) for Intel Galileo setting the appropriate jumper



Intel Galileo Board Gen 2

Working with Intel Galileo Gen 2

Working with Yocto OS locally

- It is a command line user interface:

```
telnet 192.168.10.200 — telnet — telnet — 69x12
~ telnet 192.168.10.200
Trying 192.168.10.200...
Connected to 192.168.10.200.
Escape character is '^]'.

Poky 9.0.2 (Yocto Project 1.4 Reference Distro) 1.4.2 clanton
sh-4.2#
```

- Browsing to the file system of Yocto: basic linux commands you can find here <https://diyhacking.com/linux-commands-for-beginners/>
- It includes a text editor “**vi**” and “**nano**” (use to edit your local files)
- Compiling C or C++ application files via “**gcc**” or “**g++**” compilers example:

```
compile: "gcc HelloWorld_application.c -o HelloWorld_executable_filename"
run     : "./HelloWorld_executable_filename"
```

use vi: <https://www.washington.edu/computing/unix/vi.html>

use nano: <https://www.howtogeek.com/howto/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/>

compile and run c/c++ applications: <http://pages.cs.wisc.edu/~beechung/ref/gcc-intro.html>

Working with Intel Galileo Gen 2

Working with Yocto OS remotely

Connecting to a remote machine (Galileo Board)

access the machine by executing the `ssh` command in your local shell.

For example, if `1.2.3.4` is the IP address of your remote machine, you can create an SSH connection to it using this command:

```
ssh root@1.2.3.4
```

The first time you connect to the remote machine, `ssh` will ask your permission to put the fingerprint of the remote machine in your local `~/.ssh/known_hosts` file.

```
The authenticity of host '1.2.3.4' can't be established.  
RSA key fingerprint is 12:23:34:56:21:g3:g9:93:86:af:4r:bb:11:5d:f8:h9.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '1.2.3.4' (RSA) to the list of known hosts.
```

The first time you connect to the server you should type in `yes` to proceed.

Source: <https://semaphoreci.com/community/tutorials/getting-started-with-ssh>

Working with Intel Galileo Gen 2

Working with Yocto OS remotely

- Exchanging files between host machine (*your PC*) and remote machine (*Galileo Board*) via SSH protocol via one command “scp”

```
scp file.txt galileo_username@galileo_ip_address:/some/remote/directory
```

For more options see the link here: http://www.hypexr.org/linux_scp_help.php

Thank you