CZ3004
CE3004

Raspberry Pi (RPi) System For MDP

By: Assoc Prof Nicholas Vun

© 2015 SCE, NTU

Roles and Functions of RPi

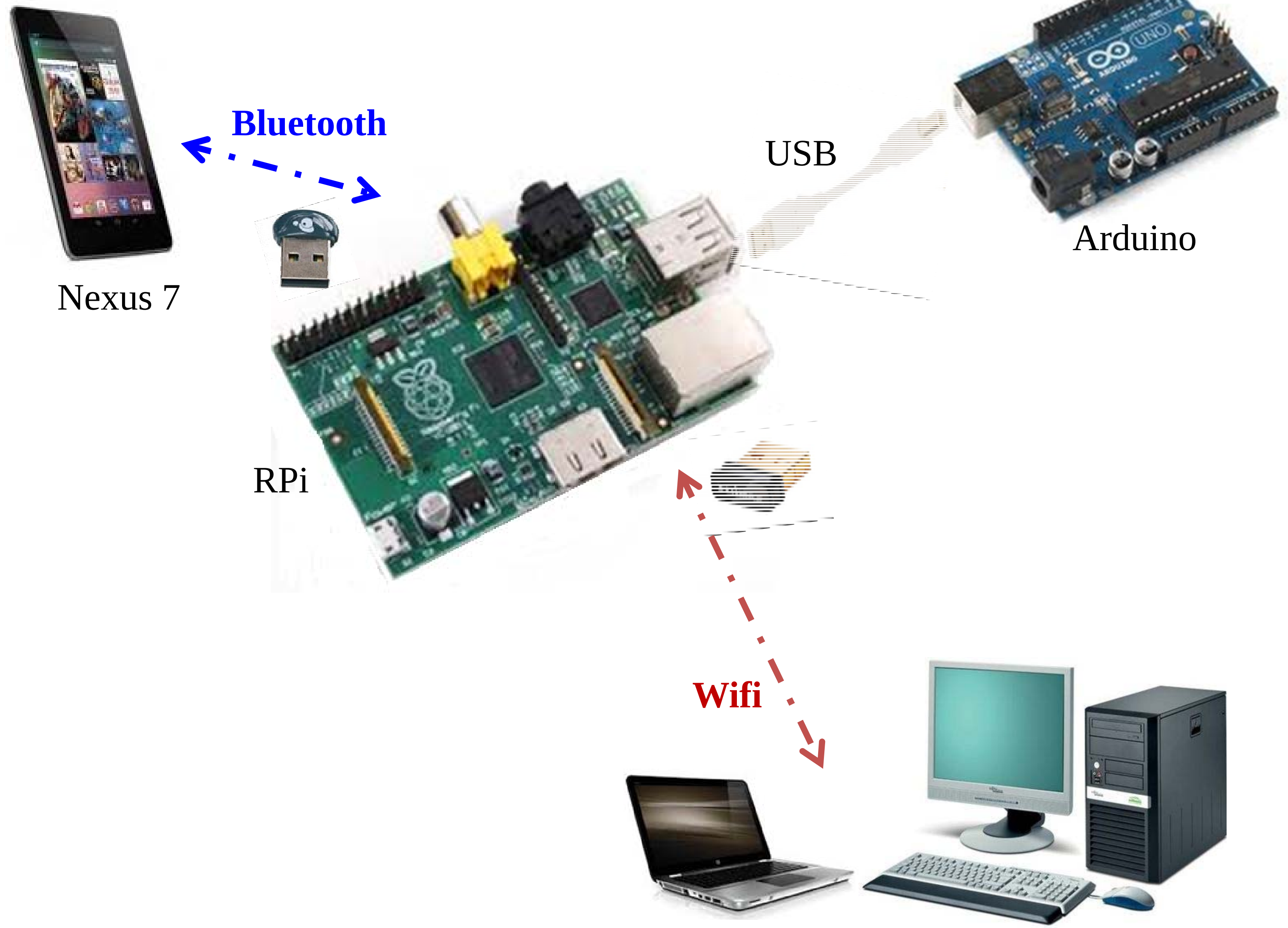
The Raspberry Pi (RPi) forms the ‘command center’ for the mobile robotic system

- coordinates and passes information between the different system modules
- but can also execute extensive functions if needed (e.g. for better real-time response)

Proposed RPi setup for MDP

- **USB host** with the Arduino
- **Bluetooth Slave** with the Nexus 7 (N7) tablet
- **Wifi Access Point** with PC/Laptop (and N7)

System Interfaces



Raspberry Pi Platform

The Raspberry Pi (RPi) board

- utilizes an ARM (ARM11) processor based SoC

  (which also contains a GPU that actually performs the boot-up of the board, as well as the ARM processor)

- runs Raspbian, which is based on Linux (Debian)

Most efficient to interact through console using text based commands (i.e. no GUI)

- so you need to know/learn certain frequently used Linux commands

- and be familiar with simple text editor

  (e.g. nano or vi)

Development System Setup

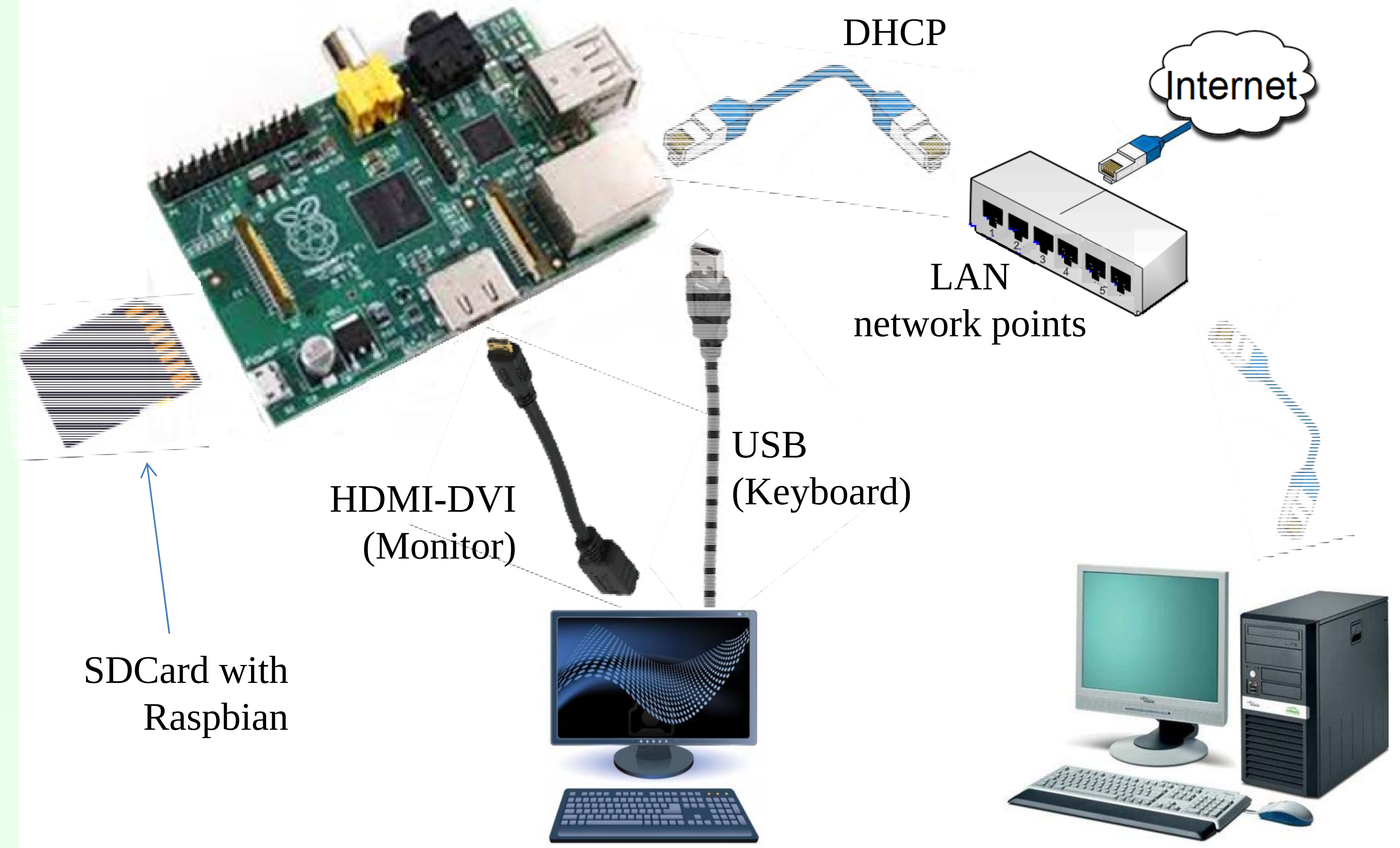
During system **development**, RPi can be setup as

- a full computing platform, with its own monitor/keyboard/mouse
- can support coding and compiling of programs directly
- connects to the internet to download and install packages as needed

The RPi platform can also be remotely accessed using Secure Shell (SSH)

- either through wired LAN or wireless (Wifi) network
- hence don't need local monitor/keyboard/mouse

Initial Setup



Monitor and
USB Keyboard

PC running SSH
(Win7 Putty)

CZ3004
CE3004

Devices Interfacing

Interfacing between the devices

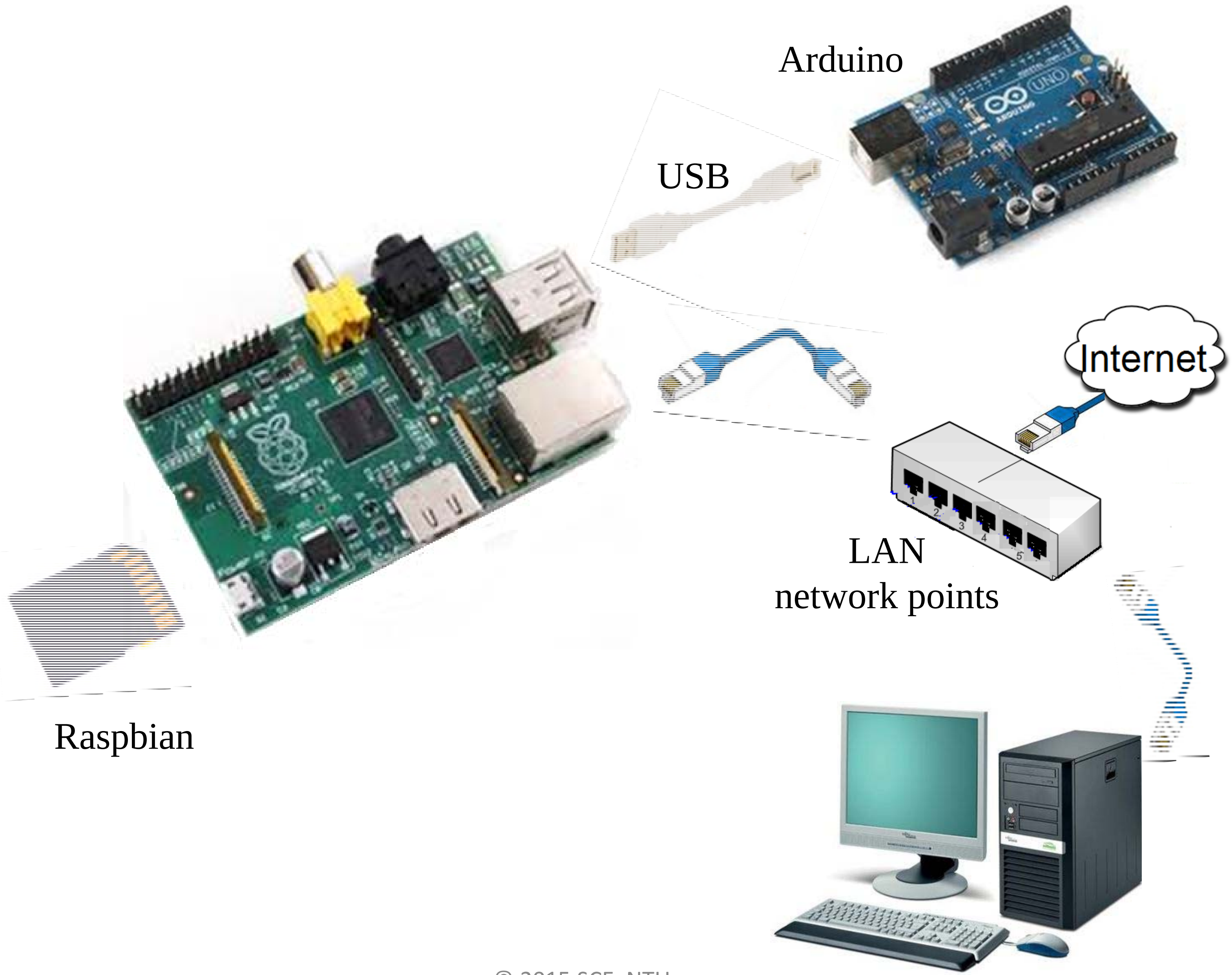
A. **RPi and Arduino**

- direct USB cable
- using ACM protocol (similar to serial port)

© 2015 SCE, NTU
(Multi-disciplinary Design Project)

CZ3004
CE3004

A - USB/Arduino Interface Setup



Devices Interfacing

Interfacing between the devices

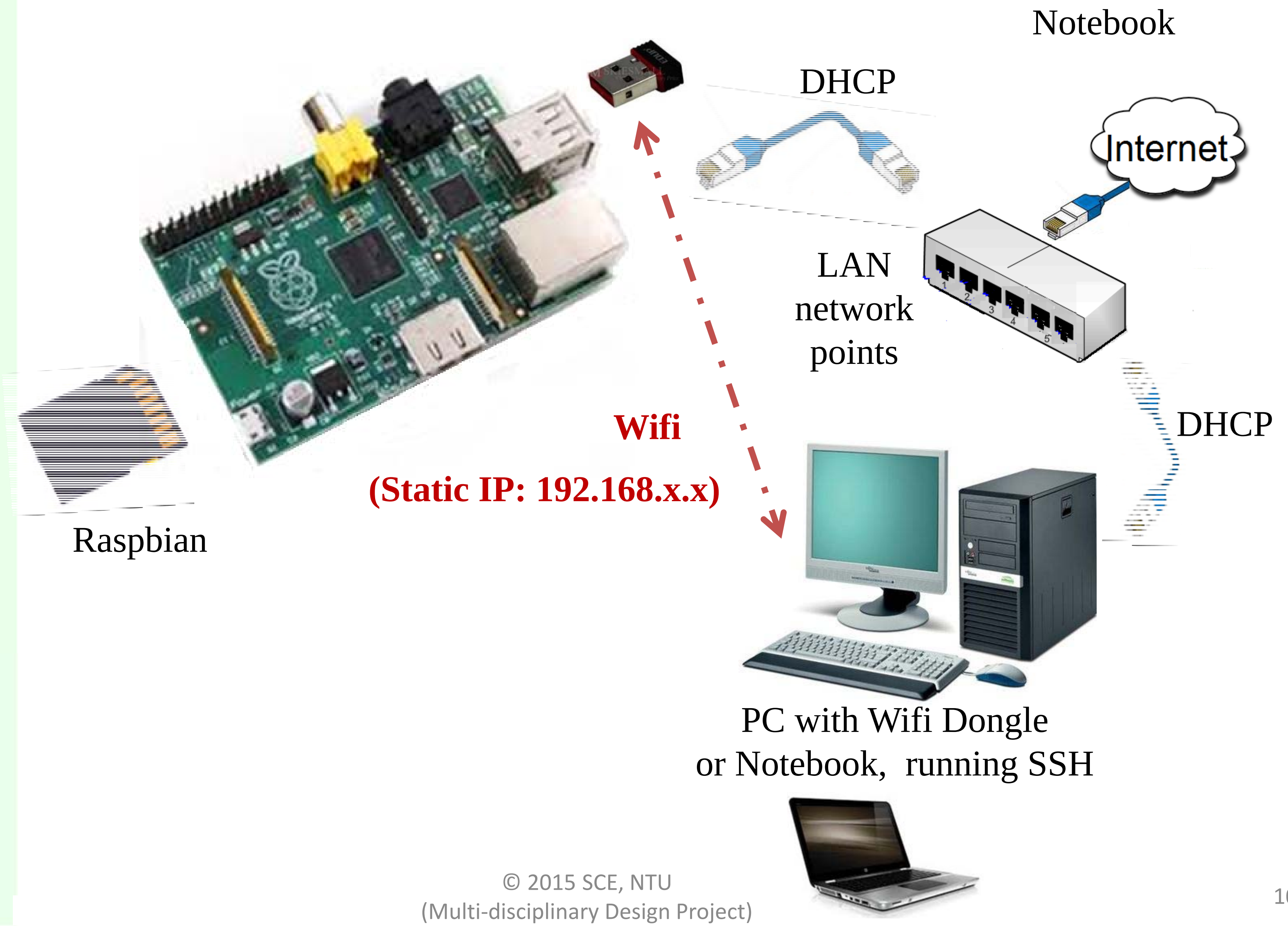
A. **RPi and Arduino**

- direct USB cable
- using ACM protocol (similar to serial port)

B. **RPi and PC/Laptop**

- Wireless IP
- Use unique static private IP
- IP socket programming

B - Wifi Access Point Setup



© 2015 SCE, NTU
(Multi-disciplinary Design Project)

Devices Interfacing

Interfacing between the devices

A. **RPi and Arduino**

– direct USB cable

– using ACM protocol (similar to serial port)

B. **RPi and PC/Laptop**

– Wireless IP

– Use unique static private IP

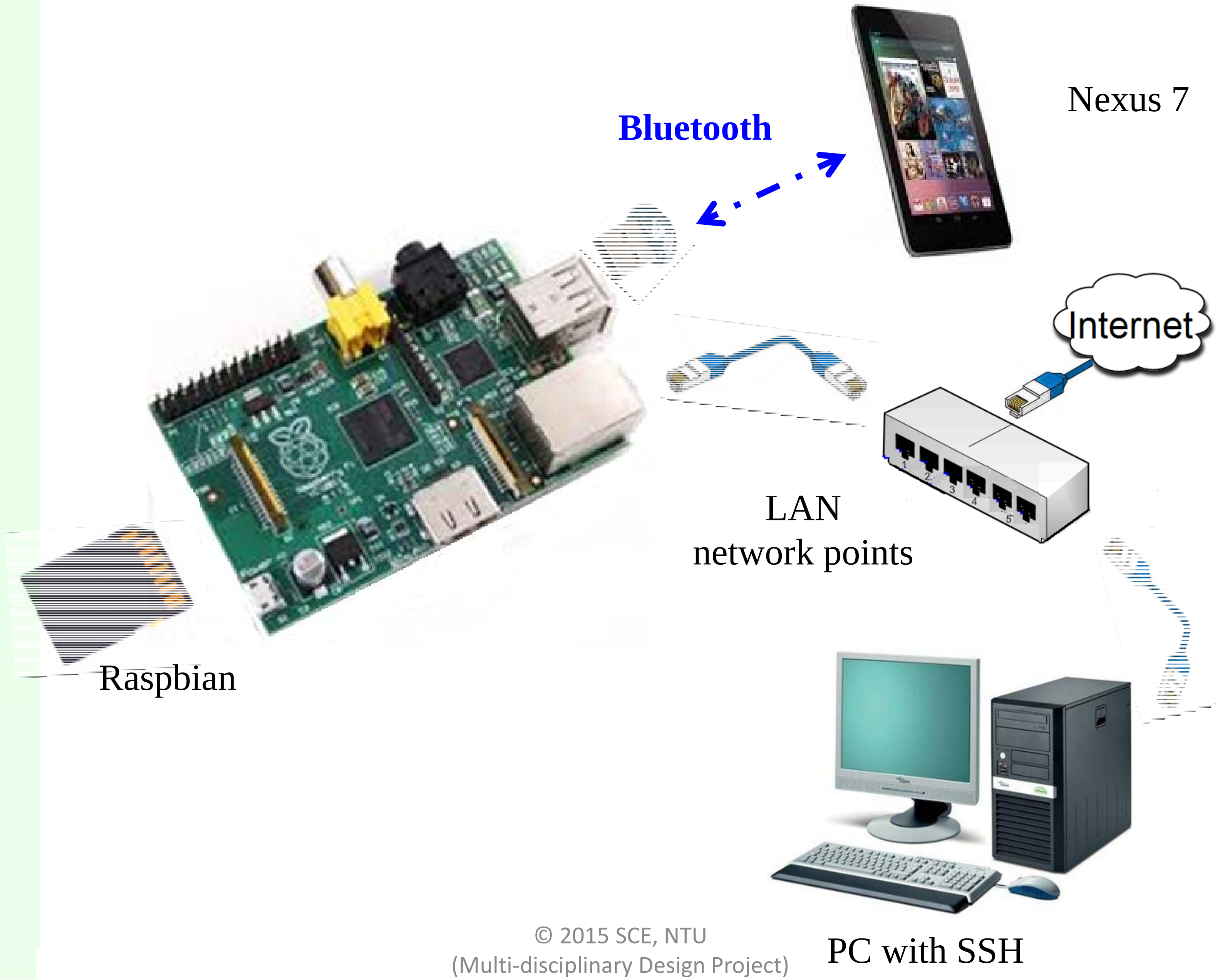
– IP socket programming

C. **RPi and Nexus 7 tablet**

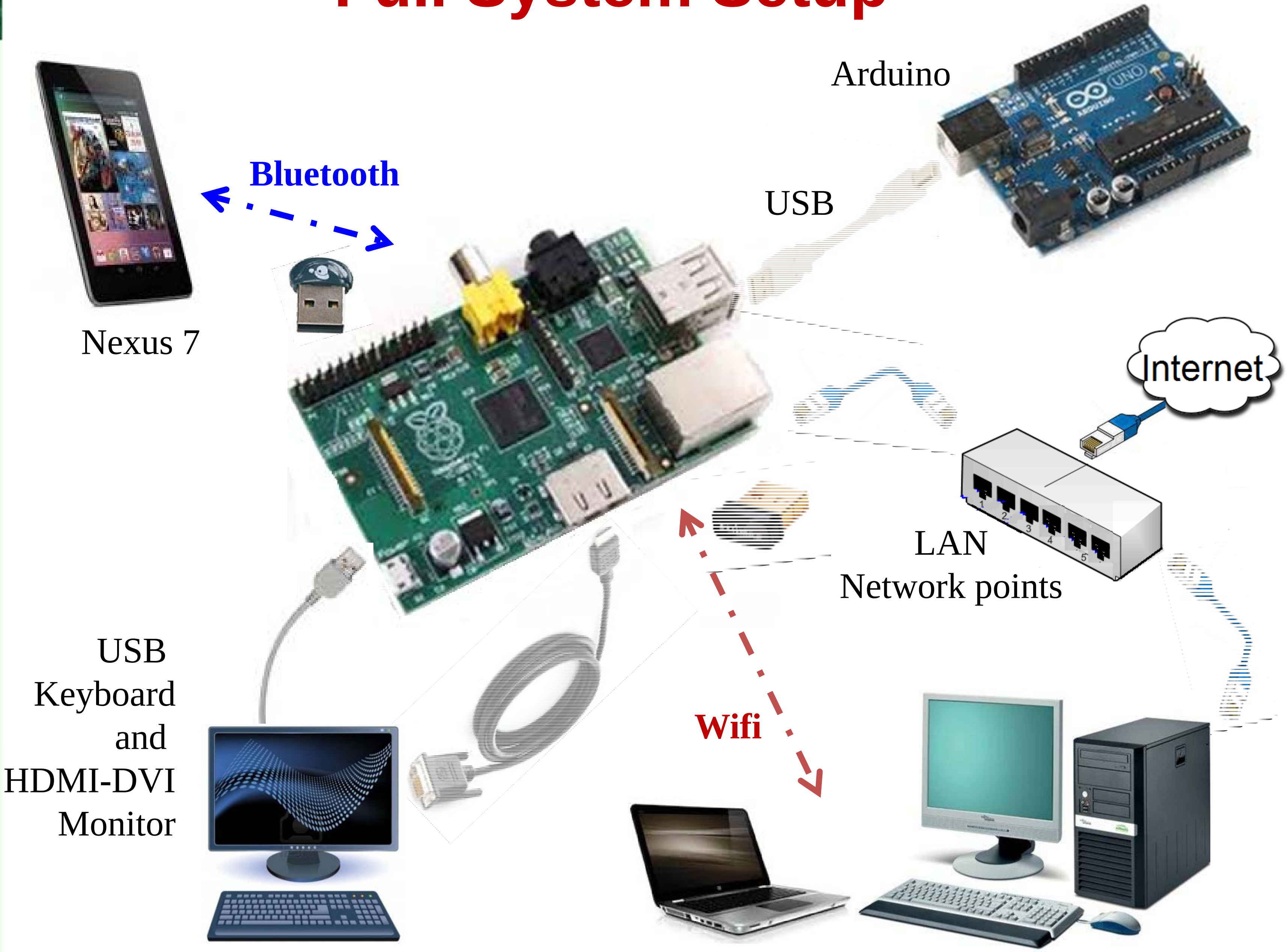
– Bluetooth wireless

– using rfcomm (similar to serial port)

C - Bluetooth Interface Setup



Full System Setup



© 2015 SCE, NTU
(Multi-disciplinary Design Project)

Guides for Platform Setup

A detailed reference document is available on NTULearn for the following setups:

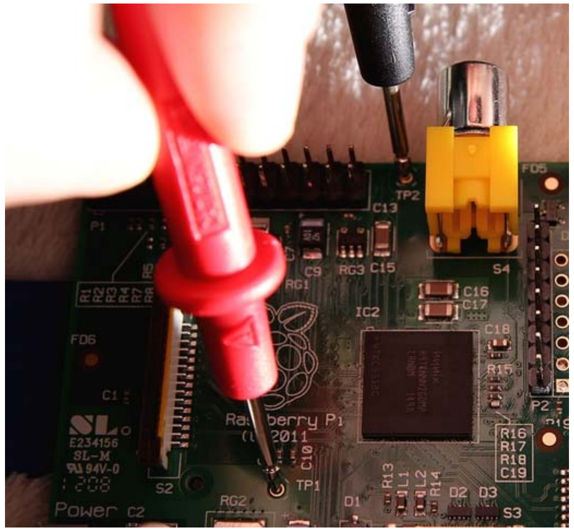
- Configure the operating system (Raspbian) for the RPi
- Remote accessing of RPi using SSH
- RPi as wireless access point, gateway through ethernet to LAN/Internet
- Bluetooth configuration to connect to N7
- Arduino USB interface

(Refer to document on NTULearn for details)

© 2015 SCE, NTU
(Multi-disciplinary Design Project)

Key things to Take Note

1. **Power Supply**
   - RPi power is limited to about 750mA due to its onboard fuse F3
   - check voltage across TP1 & TP2 if intermittent problem is observed. (Should not be lower than 4.75V)



Hence test/setup each interface <u>separately</u>.

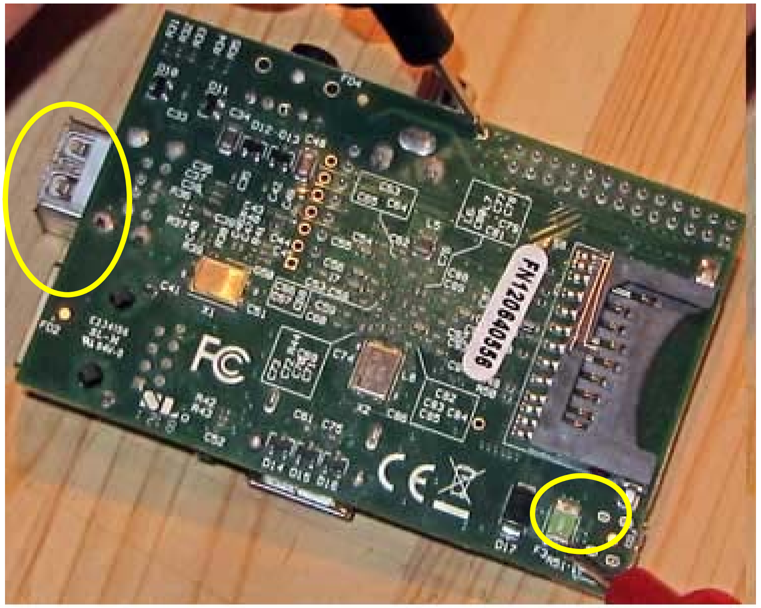
(Note: The RPi can also be back-powered through its USB port, such as when it is connected to the Arduino on the final robotic system - but then without any power protection)

© 2015 SCE, NTU
(Multi-disciplinary Design Project)

Aside: Fuse F3

Polyfuse F3

- resistance increase if current is excessive (E.g. when supplying power to all devices through RPi's USB ports)
- causes RPi voltage to drop



Key Things to Take Note (cont.)

2. **SSID and Wireless IP address**

- ensure every group uses unique SSID and different static IP range
- suggest including group number

Examples:

- SSID = MDPGrp1 for Grp #1

  :

  MDPGrp12 for Grp #12

- Private static IP: 192.168.1.1 for Grp #1

  192.168.20.20 for Grp #20

© 2015 SCE, NTU
(Multi-disciplinary Design Project)

Programming Language

Which programming language(s)?

- need to deal with real-time I/O
- need to have access to low level interfaces

Most suitable is C language

- most flexible for low level access of hardware devices and peripherals
- good real-time performance
- well support software development tool, such as gcc toolchains (included in Raspbian)

But …

Code Development

How to develop programs for RPi?

1. Code the program on PC, compile it on PC and download to RPi
   - but PC processor is x86 based, while the RPi's processor is ARM based
   - need a 'cross compiler' and toolchains
   - may not be easy to setup
2. SSH into the RPi over the network
   - login and connect to the RPi board remotely
   - use it to develop the code, and compile the code natively for the RPi processor

Summary of Interfacing Protocols

Three interfaces between the various devices

1. **RPi and Arduino**
   – direct USB cable
   – using ACM protocol (similar to serial port)
2. **RPi and PC/Laptop**
   – Wireless IP
   – IP socket programming (server-client)
3. **RPi and Nexus 7 tablet**
   – Bluetooth wireless
   – using rfcomm (similar to serial port)

© 2015 SCE, NTU
(Multi-disciplinary Design Project)

Multi-tasking Design

Three interfaces, need to execute at least three functions , each to monitor respective interface

- how should they be implemented?

i. **One program** looping through the 3 interfaces

- not efficient

ii. **Three separate programs**, multi-tasked by the OS

- but how to pass information among the programs? (Named pipe?)

iii. **One multi-thread program** (PThread), spawns three threads, one for each interface.

- easy to pass information
- may need to consider synchronization issue (Mutex, Semaphore?)

© 2015 SCE, NTU

Deliverables Checklist (For RPi)

1. The Raspberry Pi board (RPi) is able to be wirelessly accessed by PC/notebook, using the Wifi dongle provided.
2. The Raspberry Pi board (RPi) is able to wirelessly connected to the Nexus 7 tablet (N7), using the Bluetooth dongle provided.
3. The Raspberry Pi board (RPi) is able to communicate with the Arduino board through their USB ports.
4. The Raspberry Pi board (RPi) is able to execute all the above three functions at the same time, and be able to pass information among the various devices.

(Refer to MDP document for specific requirements)

Any Question?