CG3002: HARDWARE

SANGIT SASIDHAR

ELESANG@NUS.EDU.SG

E2-02-29

CONTENTS

- Human Movement Estimation
- Power Measurement
- Standard Components
- Interfacing Info
- Safety Info
- Documentation

PROJECT REQUIREMENTS

- Detect Dance Moves!!!!!
- Design your own dance moves
- Power Measurement

HUMAN MOVEMENT ESTIMATION

- What is motion capture?

- How do we do it?
  - Optical Systems
  - Non-Optical Systems

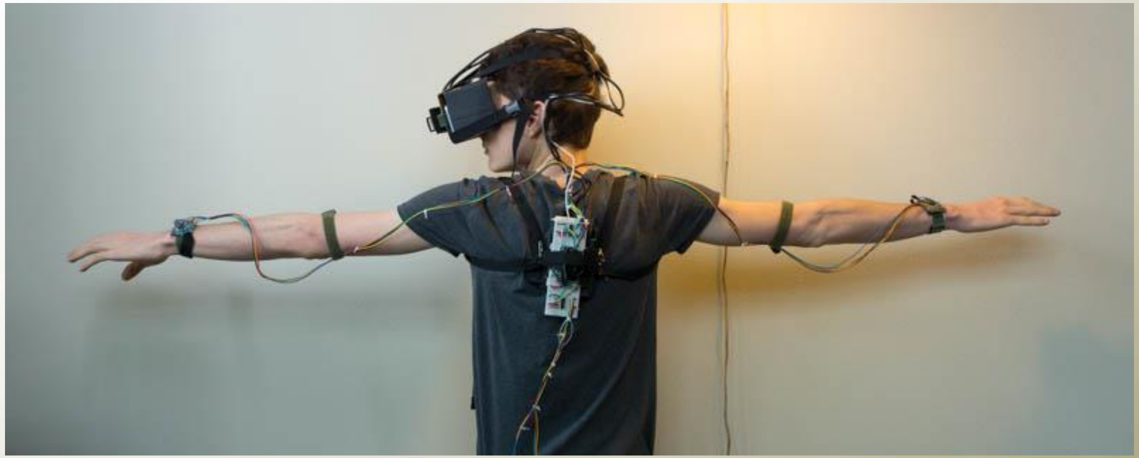
OPTICAL SYSTEMS

- Optical systems utilize data captured from image sensors to triangulate the 3D position of a subject between two or more cameras calibrated to provide overlapping projections
- Marker Based System

- Markerless System

NON-OPTICAL SYSTEMS

- Inertial Systems
  - miniature inertial sensors
  - biomechanical models
  - sensor fusion algorithms
  - $1,000 to $80,000 USD



- Mechanical motion
  - exoskeleton motion capture systems
  - $25,000 to $75,000 USD

OPTICAL VS NON-OPTICAL SYSTEMS

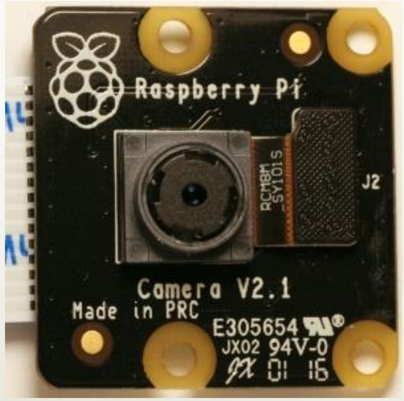
- Optical Systems
  - 3-D data
  - Processing power→ huge
  - Systems can be cheap to state of the art and very costly
  - Scaling down is difficult
- Non-Optical Systems
  - 1-D or 2D data
  - Processing Power → less
  - Systems can be cheap to state of the art and very costly
  - Scaling down is easier

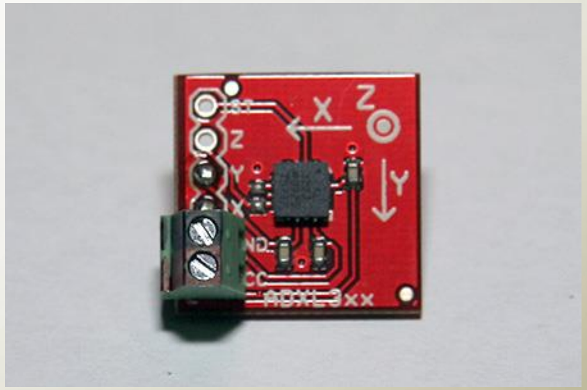
DETECT DANCE MOVES!!!

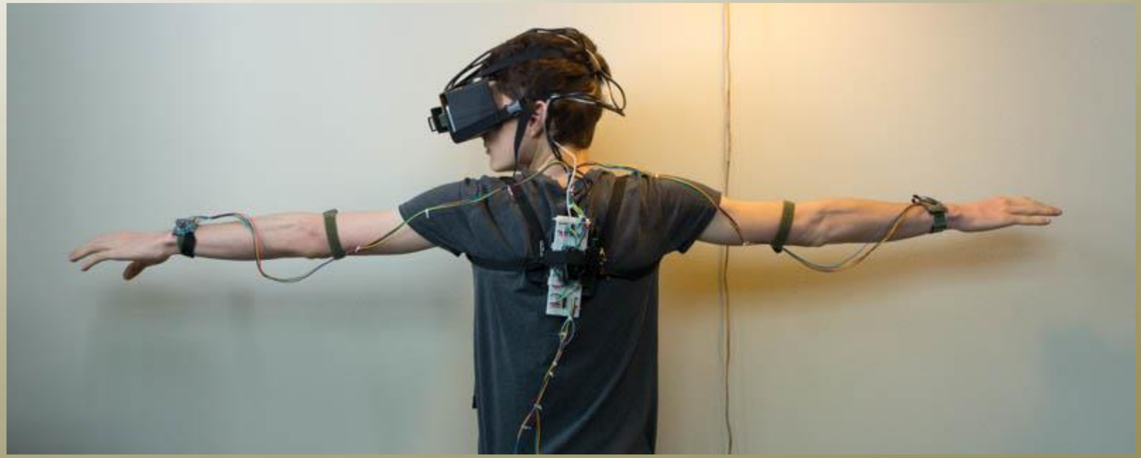
- What type of parameters need to be measured?
  - Hand movements
  - Leg movements
  - Body movements
  - Joint movements

DETECT DANCE MOVES!!!

- What type of sensors can we use?







SENSORS

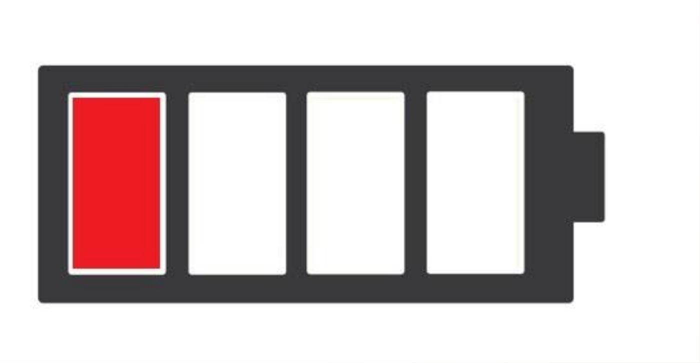
- CAMERA
- Image Processing
- Fast Processor
- Background noise
- No depth information
- Visual Occlusion

- FLEX SENSORS
- Joint angle
- Amateur processor
- Noise free data
- No other information
- 1 DOF

- IMU/ACCELEROMETER
- Acceleration, tilt, angles
- Good processor
- Some noise
- Holistic information
- 3/6/9 DOF

POWER MEASUREMENT

- Why Power Measurement?



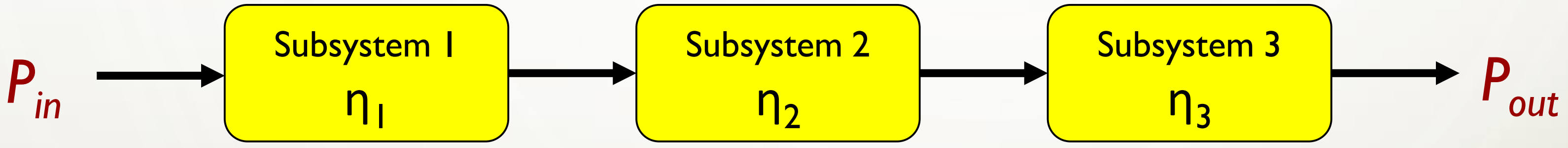
VS

VS

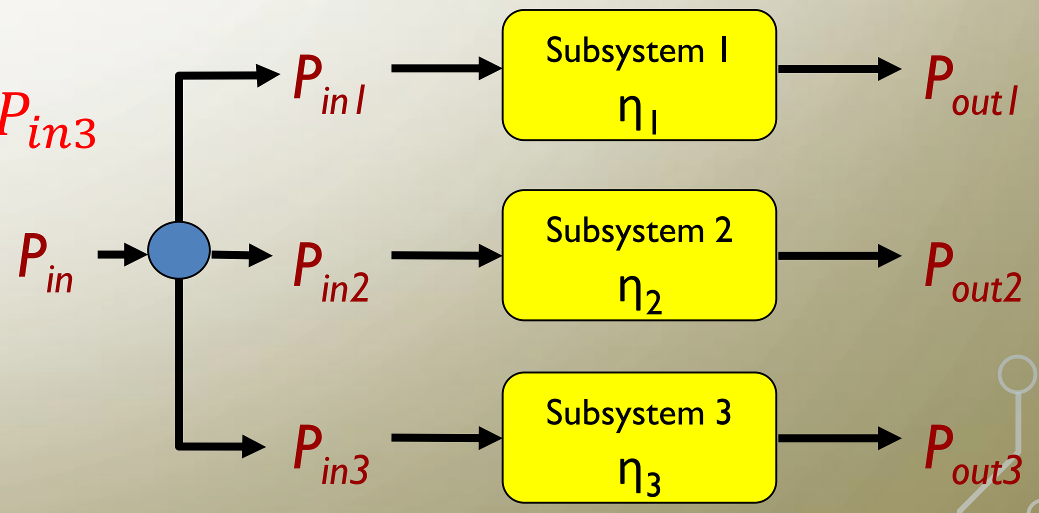
- How Big a Battery do you actually need?

POWER BUDGET AND BATTERY DESIGN

- Subsystems in Series $P_{in} = \dfrac{P_{out}}{\eta_1 . \eta_2 . \eta_3}$



- Subsystems in Parallel $P_{in} = P_{in1} + P_{in2} + P_{in3}$



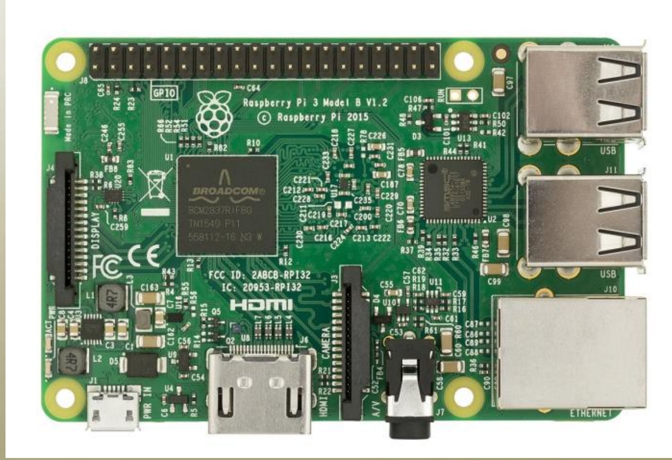
- Subsystems in Series and Parallel

LIST OF STANDARD COMPONENTS (TENTATIVE)

- 2 x Raspberry Pi 3
- Arduino Mega
- 2 x Micro-USB cables for powering RPi
- HDMI-DVI converter
- Network cable
- SD card reader
- 2 x SD cards
- Logic Level shifters (TXB0108)
- Accelerometers
- Current sensor

RASPBERRY PI 3

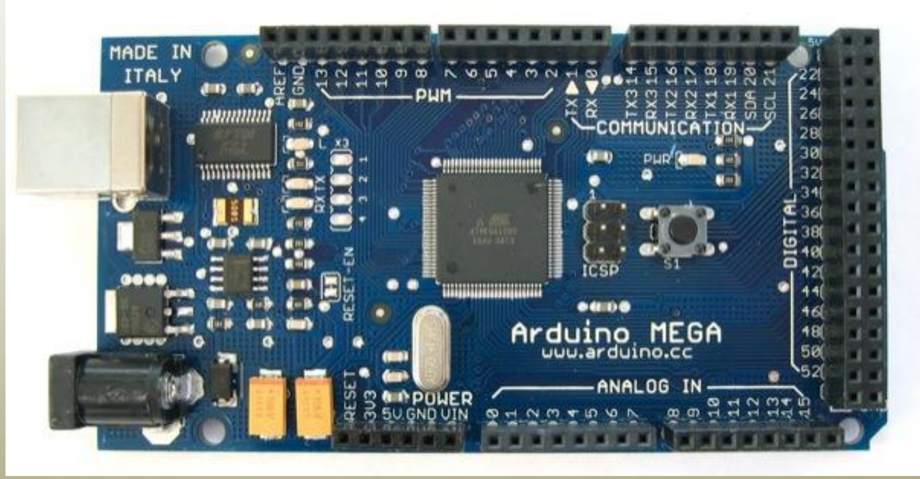
- Credit-card sized single board computer, good for applications requiring processing of data
- Raspbian OS, Python and C-libs available
- Can be accessed using monitor / KB / mouse. SSH / VNC / serial console generally used to reduce clutter
- Kernel modules for I2C, SPI etc not enabled by default in Raspbian



http://www.raspberrypi.org/

ARDUINO MEGA

- Based on ATmega1280 microcontroller, good for control-oriented tasks
- 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs, a 16 MHz crystal oscillator, a USB connection etc.
- Arduino IDE and libraries can be used for easy programming



http://arduino.cc/en/Main/arduinoBoardMega

GPIO, I2C, SPI, UART

- General Purpose Input/Output (GPIO) – configure a pin as output / input and write/read digital data to the pin

  Eg : simple devices such as ultrasound, IR, 2-line LCD display

- Inter-integrated circuit (I2C) – 2 wire, synchronous, serial, master-slave, half-duplex, in-band addressing using 7-bit addresses (+read/write bit), short-distance [Arduino Library : Wire]

  Eg : Slower devices such as accelerometer, gyro, compass etc.

- Serial peripheral Interface (SPI) – 3 shared wires + dedicated slave select, synchronous, serial, master-slave, full-duplex, short-distance

  Eg : Faster devices such as SD cards, LCD displays [Library : SPI]

- Universal asynchronous receiver/transmitter (UART, also simply called serial) – 2 non-shared wires, asynchronous, serial, full-duplex, longer distance (using a physical layer) [Library : Serial]

  Eg : Between computers / microcontrollers, RFID modules, GSM modules

INTERFACING INFO

- Take note of the voltage levels of each device (3.3V / 5V etc)
  - RPi3 operates at 3.3V, pins are not 5V tolerant. Needs a level shifter circuit to connect to 5V / TTL devices
  - Arduino operates at 5V. Needs level shifters for non-5V-tolerant devices
  - I2C level shifting requires specialized level shifters
- Take note of the max. current supplied by each pin of the device
- Read the datasheet carefully for such info
- Damaged RPi3 / Arduino will be replaced, but the cost will be deducted from your budget !!
- You have to be careful about endianness* used by the two devices being interfaced

*If **L**east significant byte occupies the **L**owest address, it's **L**ittle-endian

IMPORTANT SAFETY INFO

- Follow all the lab safety rules (non-negotiable)
- Be responsible for your own safety as well as that of people around you
- Always use a regulated, fused power source to power your project
- You need a battery to power the final system (for it to be wearable)
  - The battery specs would depend on the components you are using
  - If possible, try and use NiMH battery packs
  - Be VERY VERY careful with Li-ion / LiPo batteries
    - They can explode if you overcharge, charge faster than the recommended rate/current, charge using a charger not designed for it, draw excessive current, subject it to mechanical stress
    - Use it only if you are very careful and confident. Never charge unattended or attempt disassembly

DOCUMENTATION

- A proper documentation is very useful in debugging
- Document everything in a wiki (NUS wiki)
    - Include the links. Always have wiki open whenever you google
    - Save all the datasheets, libraries you used (you need to specify the library source in your code clearly)
- Will help you in final documentation. It will also serve as a learning journal
- "Oh, I had seen it somewhere, can't recall where" issues can be minimized
- CG 3002 Wiki (The link will be up by this weekend)

DESIGN REPORT!!! (WEEK 4)

- Do a through design – system level as well as hardware level
- For hardware, have a very clear plan
  - Block level schematic
  - Power Budget and Battery Design Calculations
  - Clear identification of part numbers, including other components to get it to work, such as resistors
  - Pin table : Which pins of RPi/Arduino will be used, how the components are connected to these pins
  - Some idea of the algorithms / libraries you would be using to get these hardware to work (or if you are writing own code, the relevant registers and configuration commands)
  - Final Product should be complete, compact, wearable – use breadboard, / veroboard or wire-wrapping / PCB (in increasing order of awesomeness). Achieve a good trade-off between cost and functionality

THANK YOU