Final Project

•Schedule, Organization
•Choosing a topic
•Example projects
•Grading
•Design Suggestions

Verilog Files

- ISE sets TABs as 3 spaces and displays correctly within ISE. However, most text viewers treat TABs as 7 spaces.
- To reformat, open the file with **emacs** and replace all TABs with three spaces using the replace-string command:

```
ESC-x                // open command line for entry
replace-string       // enter "replace-string" (emacs self completes)
TAB                  // enter TAB key
[] [] []             // press the space bar 3 times
```

save the file.

- Or in **vim** (better option, objectively), open:

```
:set tabstop=3 shiftwidth=3 expandtab
:retab
:wq
```

- Or **Sublime** (Demo):

Final Project: Schedule

- Choose project teams (email gim ASAP)
  - Teams of two or three. A single person project requires approval of lecturer.
- Project Abstract (due Fri Oct 19, submit on-line)
  - Start discussing ideas now with 6.111 staff
  - About 1 page long, a list of team members, and a one paragraph description of the project itself.
- Proposal Conference with staff mentor (by Fri, Oct 26)
  - Bring your proposal with you *and* submit on-line
- Block Diagram Conference with mentor (by Fri, Nov. 2)
  - Review major components and overall design approach
  - Specify the device components you need to acquire (*small* budget allocated for each project if component does not exist in the stock room). Get approval and will contact people to obtain the parts.

Schedule (cont'd.)

- Project Design Presentation to class (Nov 6 & 8 2:30-4p)
  - Each group will make a 10-15 min electronic presentation (~10 slides) dividing presentation among team members
  - Submit PDF on-line, will be posted on website
  - Example: F2011 Recursive Augmented Reality
  - Required attendance (3% grade)
- Project Checkoff Checklist to staff (by Nov 16)
  - Each group in discussion with mentor creates a checklist of deliverables (i.e., what we can expect each team member to demonstrate). Submit PDF on-line. Three groups:
    - Commitment – minimum goals; complexity 2x lab 4
      "Stuff we need in order to have not failed completely."
    - Expected – needed for successful project
      "Stuff we need in order to succeed"
    - Stretch goal – stands out in complexity, innovation, risks
      "Stuff we need in order to be awesome"

Schedule (cont'd.)

- Final Project Demo/Checkoff/Videotape (Dec 11 & 12)
  - Videotaped and posted on-line with your permission
- Final Project Report (Wed, Dec 13 5PM)
  - Submit PDF on-line, will be posted on website
  - Sorry, <u>no late checkoffs or reports</u> will be accepted

When Gim was in school

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering

6.711T Digital Systems Project Laboratory 28 Oct 69

Project Proposal Instructions

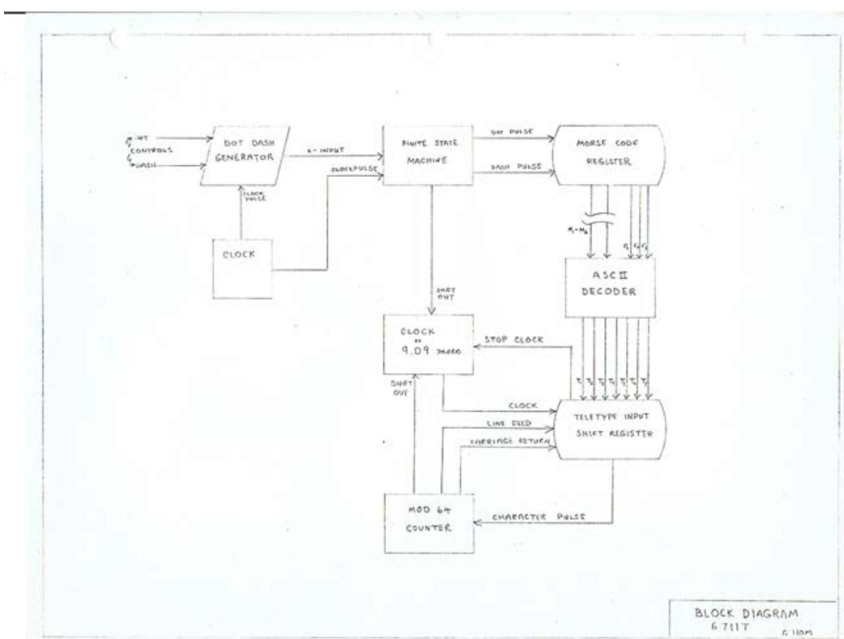
1. There should be one proposal per student.
2. The proposal itself should consist of:
   a) Cover sheet in standard fromat with a 2 to 3 typewritten paragraph description of the project. [2 copies]
   b) Block diagram attached.
   c) Estimate of amount of logic required.
3. There will ordinarily be no more than two people per project.
4. No project may be started until Labs 1 - 4 and Problem Sets 1 - 5 are completed and handed in.
5. <u>Proposals are due by the 10th of November</u>.
6. <u>Detailed logic diagrams of the entire project are due by the 1st of December</u>.
7. Actual construction of your project cannot begin until the detailed logic diagram has been approved by your T.A.! Persons not adhering to this rule will have their wiring torn down.
8. Oral presentations will be given the week of 10th-14th November.

Preliminary Oral Presentations

Every student must deliver an oral presentation of his proposed project to the instructing staff and other students. The presentation should run approximately 10-12 minutes with a subsequent 3-5 minute question and answer period. Students working as partners may not give talks in the same hour. Each student must attend the entire hour of talks in which his presentation is given. If interested in any other projects, feel free to attend. The sign-up schedule is posted in the lab.

The purpose of these presentations is to help make clear in your mind as well as ours what exactly you are attempting as a project. In addition constructive suggestions might be offered concerning your design or approach to the problem.

We landed on the moon three months prior to this....interesting



2018 End of Term Crunch

Sep 03	*Registration Day*	L01: Course overview. Digital abstraction, static discipline, logic families
Sep 10	L02: Combinational logic, canonical representations, simplification and synthesis	L03: Verilog hardware description languge. FPGA architectures Lab #1 checkoff
Sep 17	L04: Sequential building blocks, state and feedback, registers	L05: Finite state machines, Verilog implementation examples Lab #2 checkoff (Fri)
Sep 24	L06: Case study: video circuits	L07: System Integration, Clocking, number encoding
Oct 01	L08: Arithmetic circuits, adder, multipliers Lab #3, Checkoff (Tue)	L09: Behavioral transformations, FPGA
Oct 08	*Student Holiday*	L10: Analog building blocks (op-amps, DACs, ADCs), sampling, reconstruction, filtering Lab #4, Checkoff (Fri)
Oct 15	L11: Project kickoff; proposals and presentations	L12: Memories: on-chip, SRAM, DRAM, Flash Project abstract due Lab #5 checkoff (Fri)
Oct 22	L13: Communications Proposal Conferences Work on Project Proposal	L14: Image Processing - Let's go to Fenway! Proposal Conferences Work on Project Proposal
Oct 29	L15: VLSI and power Project Proposal	Project Block Diagram Meeting by 11/02 (Fri) by 5pm
Nov 05	Project Design Presentations (2:30-5PM room TBD) - attendance required	Project Design Presentations (2:30-5PM room TBD) - attendance required
Nov 12	Project Checklist Meeting with Staff Revised Project Proposals (if necessary) due 11/16 (Fri) by 5pm	Final project Project Checklist Meeting with Staff by 11/16 (Fri) by 5pm
Nov 19	Final project Short week	*Thanksgiving*
Nov 26	Final project integrtion and debugging - finishing touches! Two weeks remaining!	
Dec 03	Final project - finishing touches!	Final project - polishing! ...
Dec 10	Project Checkoff/Video recording Mon/Tue Return tool kits Tue	Wed project Report due 12/12@ 5PM (Wed) Tie up loose ends

Last modified on 06/02/2018

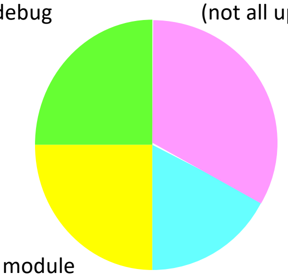
Team Organization

The Joy of teamwork.

- Most importantly, you need one
- Key decisions made jointly
  - Requirements
  - High level design
  - Schedule
  - Who will work on what, who'll take the lead
  - Response to slippage
- Lower level design exchanged for examination
  - Everyone responsible for everything
  - Design reviews tremendously helpful
    - Try it, you'll like it
- Communicate with each other early and often

Controlling Schedule

- First, you must have one
- Need verifiable milestones
- Some non-verifiable milestones
  - 90% of coding done,
  - 90% of debugging done,
  - Design complete
- Need 100% events
  - Module 100% coded,
  - Unit testing complete
- Need critical path chart
  - Know effects of slippage
  - Know what to work on when



Provide a 4-7 day contingency to deal with unforeseen issues (you'll use it all!)

Choosing a Project: Some Suggestions

- Be ambitious!
  - But choose a sequence of milestones that are increasingly ambitious (that way at least part of your project will work and you can debug features incrementally).
  - But don't expect 400Mhz operating frequencies, etc.
- It's motivating if there's something to see or hear
  - Video and graphics projects are fun (and with the labkit basic video input and output are pretty straightforward which means you can concentrate on the processing)
  - Audio/Music is low-bandwidth, so it's easy to do interesting processing in real-time (real-time is harder with video).
- Memories are often the limiting factor
  - Figure out how you'll use memory blocks early-on
- Be prepared for unpleasant surprises. Unlike the labs, there may be no solution for a particular design approach!

More Suggestions

- Be modular!
  - Figure out how test your modules incrementally (good for debugging and checkoff!)
  - Be clear about what information is passed between modules (format, timing)
- Don't be caught by the mañana principle
  - Six weeks goes by quickly: have a weekly task list.
  - How does a project run late: one day at a time!
  - Effort is not the same as progress: "Written but not tested" only means you've made a start
  - Tasks will take longer than you think
  - Final integration will uncover bugs/thinkos so test module-to-module interactions as early as you can

Design Suggestions

- Use hierarchical design
  - Partition your design into small subsystems that are easier to design and test.
  - Design each sub-system so they can be tested individually.
  - When appropriate, use Major/Minor FSMs.
- Use the same clock edge for all edge-triggered flip-flops
  - Beware of clock skew, don't gate the clock
  - If you have multiple clock domains, think very carefully about how you transfer information from one to another
- Avoid problems from 'glitches'.
  - Always assume that combinational logic glitches
  - Never drive a critical asynchronous control signal (register clock, write enable) from the output of combinational logic.
  - Ensure a stable combinational output before it is sampled by CLK.
  - When needed, create glitch-free signals by registering outputs.

Design Suggestions (cont'd.)

- Avoid tri-state bus contention by design (more next week)
- Synchronize all asynchronous signals
  - Use two back-to-back registers
- Use asynchronous memories properly (more later this week)
  - Avoid high Z address to SRAM when CE is asserted.
  - Avoid address changes when WE is true.
  - Make sure your write pulse is glitch free.
- Use care when incorporating external devices
  - Use bypass capacitors on external components to deal with noise
  - I/O pads are slow, not all signals have the same delay
- Chip-to-chip communication
  - Beware of noise (inductance)
  - Might need to synchronize signals
  - Can also use "asynchronous" protocols

Project Grading (35% Total)

- Deadlines and effort (8 %)
- Problem Definition and Relevance, Architecture, Design methodology (10%)
  - What is the problem
  - Why is it important or interesting
  - System architecture and partitioning
  - Design choices and principles used
  - Style of coding
  - All of the above should be stated in the project and report
- Functionality (8 %)
  - Did you complete what you promised (i.e., graded by the checklist)
- Complexity, Innovation, Risk (9 %)

Warning!

- Designing and simulating is easy
- Integrating into real hardware FPGA is difficult
- Plan on unexpected (expected) problems.
- Examples:
  - Works in simulation
  - Works with slower clock

Project Grading

- Functionality grading
  - It works in simulation: grade 0%
  - Unable to demo/test because my partners' module isn't working: grade 0%
- General project grading guidelines
  - approximately 2x hardest lab: grade 10-19
  - demonstrates a superior understanding to digital systems and implementing complex systems - perhaps with multiple time domains, interface to external devices, flash memory, audio, etc. 20-29
  - a top notch project that really stands outs with complexity, innovation and risk 30-35

Presentation & Report Grading (13%)

- Project Proposal (2%)
- Class Presentation   (6%)
- Final report (5% technical)

Required Attendance (3%)

- Design presentations 2:30 - 4:00p
  Tue Nov 6, Thur Nov 8

Report Grading Rubric

- For technical grading, we assign a max of 5 points as follows:
  - Technical content of overview/motivation: 0, 0.5, 1
  - Logical, readable diagrams and timing (if appropriate) 0, 0.5, 1
  - Enough details so the project can be replicated by a fellow student 0, 0.5, 1
  - Discussion on tricky circuits/challenges/measurements of interesting signals (if appropriate) 0, 0.5, 1
  - Lessons learned, advice for the future projects, 0, 0.5, 1

Choosing A Topic

- You only have 6 weeks total (once your proposal abstract is turned in) to do this project.
  - It is important to complete your project.
  - It is very difficult to receive an "A" in the class without having something working for the final project.
- The complexity for each team member should 3 times the complexity of the lab assignments.
- Some projects include analog building blocks or mechanical assemblies (infrared, wireless, motors, etc.). However, keep in mind that this is a digital design class and your design will be evaluated on its digital design aspects.
- Complexity, risk and innovation factor.
  - We will give credit to innovative applications, design approaches
  - More complex is not necessarily better
- Look through previous projects for inspiration (see website)

Sample Projects

- Live Action Mario Kart
  - Brad Gross, Jono Matthews, Nate Rodman
- Conductor Hero
  - Natalie Cheung, Ned Twigg, Yuta Kuboyama
- Digital Sonar
  - Zhen Li, Bryan Morrissey, Brian Wong
- A Hardware-based Image Perspective Correction System
  - Matthew Hollands, Patrick Yang
- Self Parking Car
  - Kevin Hsiue, Frank Ni

Final Project Ideas

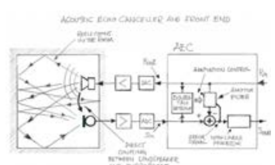
- **Gadgets, digital systems**
  - FPGA Function Generator with laser display
  - Multimeter with voice output
  - Analog Voltmeter
  - FPGA Fitbit
  - Virtual pool with sound
  - Remote control hand movement
  - Virtual golf
  - Camera based arcade game
  - Motion tracker alarm system

- **Graphics/Video**
  - Star Wars Virtual Light Saber
  - 3D fly by
  - Movement tracker/playback
  - Real time animation with camera
  - Airplane console
  - Wire frame editor/display
  - Camera with green screen
  - Virtual postcard
  - 3D display (two cameras – tough!)
  - Automatic keystone correction
  - Softball
- Audio, music, lighting
  - Music synthesizer
  - FPGA phone system
  - DJ Control system
  - Light panel control system
  - Virtual surround sound
  - Time stretching audio or Time domain harmonic scaling (not for faint of heart)

Random Project Ideas (What I'd do if I were young)

Margaret Hamilton, Director of Software Engineering Apollo Program, 1967

- Rebuild Apollo Guidance Computer (AGC):
  - Source code here: https://qz.com/726338/the-code-that-took-america-to-the-moon-was-just-published-to-github-and-its-like-a-1960s-time-capsule/
  - Schematics here: http://klabs.org/history/ech/agc_schematics
  - Real one was implemented using nothing but 3-input NOR Gates !
- Implement a Neural-Net/Some kind of Assisted/Unassisted learning:
  - FPGAs have benefit of parallelizability
  - Some decent repos out there to get started/serve as inspiration
  - Binary Neural nets (avoid using bloated float32's!)
  - Classify some things without the crutch of TensorFlow, numpy, etc…
- Noise-Cancellation using DSP:
  - Don't need to rely on feedback only of to get 6dB of suppression
  - FFTs and cut out the noise



- Acoustic Echo Cancelation:
  - Have the FPGA "learn" the room's acoustics so it cancel out its contribution and listen for sounds it isn't producing (like an Amazon Echo)
- Implement Ethernet (up to 100 Mbit/s should be doable on our equipment) (CRC32,

https://en.wikipedia.org/wiki/Margaret_Hamilton_(scientist)
https://www.vocal.com/echo-cancellation/acoustic-echo-canceller/

Electronic Design Studio (EDS)

Laser cuter – PSL6.75 with 75W laser
Part size: 32" x 18" x 9"
Cut by "printing"

Bridgeport Mill!

- Any fabrication needs, soldering, random parts, we can get from there. If John isn't in, find Dave nearby or Joe in his office nearby (38-583)

Parts of Interest

(if time permits)

Where we were…where we're going

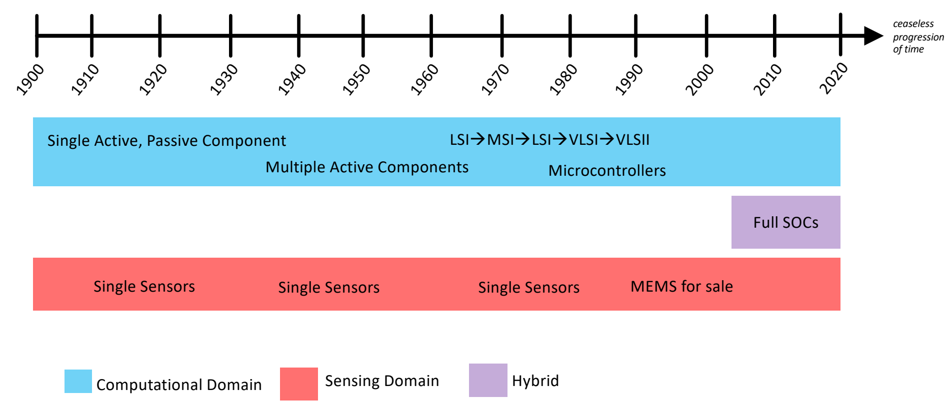
- Everyone knows and loves to misquote Moore's Law
- Computational Power has indeed grown exponentially in the last half-century
- Along with that, lurking in the depths, has also been a remarkable development of fully-integrated systems that we've really only started to see break out in the last ~15 years.

https://en.wikipedia.org/wiki/Moore%27s_law

Case Study (Analog): High-voltage Amplification

- APEX PA3440 High-voltage Operational Amplifier:
  - 340 VDC supply limit
  - 120 mA output current
- Where before you needed to design an entire circuit and worry about capacitive loading, death, you now can buy this for $11.00 and drive a Piezo pretty easily

What was the complexity of a single part that you could buy?



Integration Provides Robustness

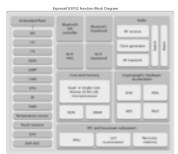
- Much of 20th century focused on integration at the electrical/computational level
- Last few decades saw further integration in with sensors, actuators, communication modules, etc…
- Emergence of real SOCs

German VCL11, 1938: two tubes in one tube!

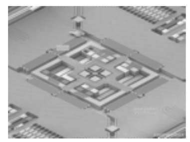
1950s Op Amp from George Philbrick Assoc (Boston)

7400 SSI Integrated Circuit

8080 MCU



ESP32 SOC Functionality Blocks



iPhone 4 (gyro)

http://eecatalog.com/sensors/2012/05/24/mems-trends-smaller-cheaper-everywhere/

Case Study (MEMS SOC): Accelerometers:

1950s Soviet Accelerometer

ADXL335

MMA8452Q

- First MEMS accelerometer: late 70s
- ADXL50: (Analog Devices 1991) (single axis in 25mm^2)
- ADXL335: Analog Out Accelerometer (1998)
  - Voltages for three axes of acceleration
- MMA8452Q: Digital Accelerometer (mid 2000's)
- MPU925X series (or LSM9DS1) series (last few years):
  - 9 DOF (Accel, gyro, compass)
  - Onboard sensor-fusion, orientation integration, DSP engine
  - In-built pedometry, cycle motion detection, wake-on-disturbance etc…

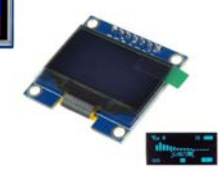
MPU9250

Parts of Interest

Small Displays:

Random TFT off of Amazon

- TFTs Thin-Film Transistors:
  - color
- OLEDs (Organic Light-Emitting Diodes):
  - Faster
  - Better viewing angle, colors, efficiency
- If you want to animate anything, if possible get an SPI interface:
  - i2C is too slow to give sufficiently high speed
- Don't forget computer monitor (is a small display core to your project?)



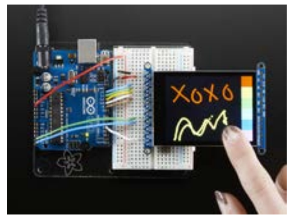
Cheap Monochrome OLED using SPI

Color OLED from Adafruit

Touch Screens

- Resistive:
  - Accessed using analog measurements (sometimes digital readout)
  - Usually only one point of contact allowed
  - Generally not good..those early android tablets with them were borderline unusable
- Capacitive:
  - Accessed via i2c, SPI registers
  - More advanced chipsets have in-built gesture detection (swipes, etc..) in addition to touch-points
  - Actually work

Random-resistive Touchscreen from Amazon



FT6X06-family based touchscreen

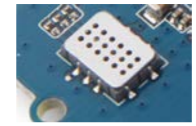
Environmental Sensors

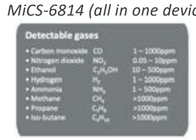
- Lots of single-chip environmental sensors available:
  - UV
  - Temp/Pressure/Humidty
  - Smoke
  - Gasses
  - Particulate Matter
- Single-chip spectrometers…probably overkill for us (do RGB extraction from camera pixels), but want to show it since it is cool
- Hamamatsu C12880MA
- ($400, but we can dream)


CCS811 Gas sensor
BME280 pressure/temp/humidity


GUVA-S12SD UV Sensor


TCS34725 RGB sensor


MiCS-6814 (all in one device!):




C12880MA

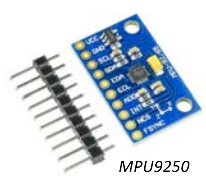
Adafruit.com

https://www.tindie.com/products/onehorse/air-quality-sensors/

Accelerometers/IMUs

- Both digital and analog models available. Really cheap
- Orientation Determination (using gravity and compass)
- Relative Motion Determination


ADXL335


MPU9250

We use in Lab 5

Time-of-Flight Sensors

- Cheap, shorter-range LiDAR (couple meters max)
- VL53L0X by STMicro
- ST has the big patents in this field
- Used currently in proximity detection on phones, etc…
- ~$10.00
- ~1 cm resolution (10 picosecond difference in light return)


VL53L0X


Breakout from Pololu

DDS Modules

- Function Generators on a chip
- AD9850 DDS: Direct Digital Synthesis:
- Can create sine waves, square waves from 0 to 40MHz via digital control very quickly
- Phase-shift control as well!
- AD9833: Slightly different version of AD9850, SPI controlled, capable of 0.004 Hz resolution from 0 to 500 kHz with right settings/0.1 Hz resolution up to 40 MHz

AD9850 Dev Board

AD9833 Dev Board

Cameras

- OV7670 ish Series:
  - There should be some basic modules that already work with this
- Other variants exist OV2640, *should* be able to go faster…looking at communication with it now.
- Tons of others

OV7670

MT9M001 Infrared Camera

OV2640 (w SPI interface)

Raw OV2640 (should be able to get 60fps out of this with subsampling)

OV2640 (w SPI interface)

Front chip is a video frame buffer

Altera CPLD (like FPGA)

Specialty Imagers

- Line Scan Devices:
  - 128 x 1 pixel array
  - Spits out a 128-long pixel array really fast
- Motion/Thermal Detection Devices
  - Near/Far Field motion/body presence

Line Scan Camera

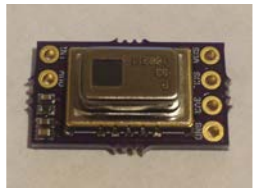
https://www.tindie.com/products/AP_tech/tsl1401cl-linescan-camera-/

TPiS 1S 1385 / 5029

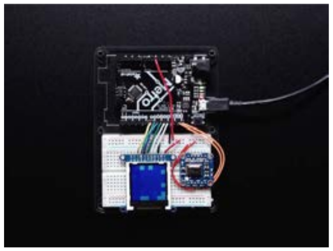
https://www.tindie.com/products/onehorse/calipile-tpis1s1385-presence-and-motion-detector/

AMG8833 Thermal Imager

- 8x8 thermal sensor array
- Effectively a Low-res camera that works off of thermal intensity
- $40 currently

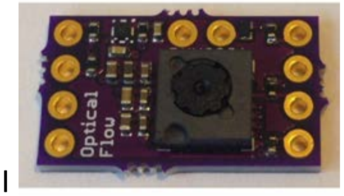


AMG8833 (from tindie)



Optical Flow Sensors

- Basically like long-range optical mice
  - (many feet with a lens)
- PWM3901 is a good one, uses SPI communication and can make/give measurements at ~100 Hz
- Great because they provide an convenient means of dead reckoning with sub-inch resolution which cannot be fixed using IMU math or current GPS



https://www.tindie.com/products/onehorse/pmw3901-optical-flow-sensor/

Board with combined PMW3901 and VL53L0X (optical flow and distance)

Wireless Communication

- Bluetooth: HC05 modules are designed to take in/put out UART (serial like from Lab02B) and chipset handles conversions (good up to ~50 ft or more with LOS)
- Infrared is reliable over short distances…(need LOS)
- Nordic Semiconductors makes some devices that have flooded the market with their own brand of 2.4GHz brand communication (popular from Arduino community)
- 433 MHz, 915 MHz ISM band communications

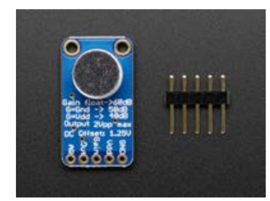
HC-04/5/6 Bluetooth series

Infrared TX/RX pairs (from Lab 5)

NRF24L01

Microphones

- Electret and MEMS capacitive microphones are standard now
- Either use analog out OR models that digitize on-chip (usually convert into I2S which is a standard audio-information digital communication protocol…we'll discuss next Tuesday)



Electret with selectable gain (Adafruit)

ICS43434 Digital Output Microphone (Invensense), uses i2s protocol

https://www.tindie.com/products/onehorse/ics43434-i2s-digital-microphone/

Audio Out

- Analog or i2s amplifiers, DACs, etc… are available.
- Most DACs have i2s inputs, and many modern electronic devices convert audio information immediately to/from i2s
- If you're considering an audio project might be a neat protocol to look into (although the AC97-setup on the older labkits works well too) (Vivado has an i2s core)

MAX98357A-based board from Adafruit: 3W Class D audio amp i2s input

Computer Interfacing

- We've already explored this in Lab 2B with USB-to-RS232 interfacing
- Nexys4 supports USB slave mode (haven't tested drivers yet)
- Also have a collection of FTDI converters (convert USB to/from UART (serial-readable)
- **Use tools like these to help put test cases, example data, record outputs, etc…**

FTDI converter
(Broken out USB to UART)

Things to Consider

- Level Shifting?:
  - Nexys4 DDR runs at 3.3V
  - Some sensors run at 5V
  - Others run at 2.7V
  - Check! May need to use level shifters and these add their own delay/skew to signals so be aware with high-speed data transfers
- When buying, check the datasheets. For every well-documented version of something there are "clones"/variants/obscure chipsets that are also sold and are often cheaper.
- Your project should be complex but do you want to focus on the complexity of a sensor or a protocol or a higher level?
- FPGAs are neat* to develop with because there's generally not a **`do_everything()`** like you'll see when working in regular Arduino-family devices. Make lemons into lemonade…more opportunity to read data sheets!

*also frustrating