3 Waldo

A Waldo is an input device used to track a motion that can then be used to move another device (output) that is typically similar in structure (though not necessarily in size) to the input device. If you talk to anyone you can call it master-slave teleoperation or remote manipulation so we don't get sued, http://www.character-shop.com/waldo.html, but Waldo is more fun to say.

This isn't just for fun in animatronics. It has uses in dangerous or challenging environments:

https://www.youtube.com/watch?v=Jd91KWzyrsY
https://www.youtube.com/watch?v=7YEMMpngZTE
https://www.youtube.com/watch?v=eOAKX5oAVMg
https://www.youtube.com/watch?v=pv8UrMUOkww
You will be creating a Waldo that has at least 2 degrees-of-freedom (DOF). This means it will have at least two joints which can be controlled independently. Extra credit for 3 or more DOF.

3.1 Waldo input

Outcomes: A good understanding of how to use the ADC on the ATMEGA32U4, and some experience with mechanical design and interfacing with an analog sensor. .

Time expectation: medium .

You will be making two similar structures (one input, one output), one that you move with your hands that the system senses, and the second structure that is driven by motors copying the motions of the first structure. In this first part of the lab you will be building the input (sensing) portion of the Waldo.

We have reserved the fabrication staff to lasercut and limited 3D printers parts for you, so the designs need to be finished and submitted early in the process (3.1.1).

3.1.1a Plan out what you are going to build. Think about how it is going to move and how to attach the sensors and servos. It may be a good idea to see how the servo in your kit can mount to the inside of your structure and the limits of the motion. **Create and submit a dimensioned drawing of both input and output sides of your waldo system. A CAD drawing will work well here, but is not required. Pay attention to mounting details. Bring to a meeting with TA during recitation to be sure that your design is feasible and get recommendations.**

3.1.1b Create files that can be used to fabricate the components using the laser cutter and or 3D printers available in the RPL. **Submit the required files to be fabricated on canvas as separate files. Be sure to have your parts have your name etched in them so we know who they go to. On canvas, also submit a list (with one sentence description) of the parts you have requested to be made.**

3.1.2 Now you need a way of reading ADC channels continuously. Create two subroutines: one for setting up an ADC port (so the user can specify what port they would like to setup), and one for reading the setup ADC port. Try to make your subroutine as clean and efficient (short and fast) as possible. We will post the routine we like best as an example for the rest of the class. Optional: write the routines (or add more) that will allow the efficient reading of multiple ADC ports. **Submit your commented code (we will have a separate submission for code this time).**

3.1.3 Interface the Teensy with two analog sensors (such as potentiometers) so their positions can be displayed to the USB serial port. Build the sensing side of your Waldo and submit a video of it moving with the position of the joints being moved and the angle being displayed in a serial window on a computer. **Submit a video (as a link in your document: youtube, google drive, etc.) of your device and the serial window showing the motion, include drawings, and any code used (we will have a separate submission for code this time). Provide a short discussion of the sensitivity of your device (the number of ADC counts over the full range of motion, linearity and apparent noise sources.)**

3.2 Waldo output

Outcomes: A good understanding of how to use hobby RC servos and control them with a microcontroller. Also creating mechanical joints that interface with servos. .

Time expectation: small .

3.2.1 Finalized hardware design

Finalize your design for the output side of your waldo. **Include dimensioned drawings of the final version of your mechanical design (they may be the same or different from what you submit in 3.1). Include a short description of your mechanical design, the design choices you made, and its intended movement and functionality (exceptional mechanical design and/or creativity will be rewarded).**

3.2.2 Hardware fabrication and analysis

Make the output side of your waldo. Use the supplied RC servos, or you may optionally purchase your own servos. Do an analysis of the potential current usage by your circuit, including the current sourcing capabilities of your power supply. You may need to search online for more precise SG90 servo current specifications. **Submit a document that includes your analysis of the total current draw of your circuit in the worst case (teensy, potentiometers, servos etc.) as well as the current sourcing capability of your power supply(s). If you use something other than the SG90 servos, include a spec sheet for the servo you used. Include a schematic circuit diagram of both the input and output sides of your waldo that includes the power source(s).**

3.2.3 Software Integration and demonstration

Write code that integrates everything to make a complete master-slave system. You may optionally use the events and services model, though this exercise is relatively small. Once completed, your output should accurately mimic the input with minimal delay. Look at the example videos provided for reference of expectations. **Submit well commented code in a separate code submission section on Canvas. Get checked off by a TA showing the system working. Show the full range of motion of each degree of freedom and also the input and output matching at some intermediate points.**

3.2.4 Do a dance!

Show us what your waldo can do! **Submit a video of your robot dancing to a 10 second portion of music choreographed to "Spanish Flea" by Herb Alpert & the Tijuana Brass.**

https://www.youtube.com/watch?v=hxsOXOPni0o **Submit a link to your 10 second video.**

This section is not graded. It is used to inform the teaching staff for future labs.

3.3.0 Briefly submit an estimate of the hours you spent on each section of this lab. Indicate any troubling spots (areas that could be improved in instruction) or other thoughts you may have about this lab or logistics for the course thus far.

Lab submission instructions

Please follow the instructions below while submitting your answers for the labs:

- Always submit your PDF write up separately from zipped files
- CAD and SW files
  - Name the file as "`YOURLASTNAME_YOURFIRSTNAME_PARTNAME`" Eg: "`DOE_JOHN_LEFTARM.DWG`"
- Text answers & all diagrams, pictures:
  - PDF file with text, diagrams, pictures, and links to videos
  - Label your diagrams and pictures clearly
  - Name the PDF file as "`YOURLASTNAME_YOURFIRSTNAME_LABNUMBER`" Eg: "`DOE_LAB2.3-2.4`"
  - Submit your PDF file to canvas directly always, do *not* include it in a .zip file
- Code:
  - Make sure you comment your code nicely and concisely, delete commented out lines for your submission
  - If the code is small (less than ~10 lines), paste the code into your PDF submission
  - If it is bigger than 10 lines:
    - Zip the entire folder including the Makefile, "`src`" files and header files
    - Label them intuitively (Eg. for 3.1 by someone with last name Doe, name the zip file "`DOE_Lab_3.1`")
    - Mention clearly what zip file corresponds to what code in your PDF submission, submit each zip file to the code submission page
    - Each submitted code file should be able to be downloaded quickly and separately from all other files
- Video:
  - Upload onto some web platform (YouTube/Google drive etc.), ensure sharing preferences are appropriate and anyone who has access to the link will be able to view the videos
  - Include the links in the PDF file submission
  - Please do not upload the video on Canvas, even in a .zip file (might be too large to download)
- Please do not put zipped folders inside zipped folders