Lott/Jayaraman Treadmill Assembly Guide

Principal Scientist

Vivek Jayaraman

Group Leader, JFRC

Email: vivek@janelia.hhmi.org

Principal Engineer

Gus K Lott III

Neurobiological Instrumentation Engineer, JFRC

Email: lottg@janelia.hhmi.org

Abstract

This document acts as a total assembly guide to get you from zero to a fully functional optical flow tracking system. It describes the included documentation package, provides a full parts list for purchasing, and an assembly and test guide to guide from raw parts to Matlab data acquisition while also providing information for advanced users who may want to modify design files.

Treadmill System as described in:

Seelig JD*, Chiappe ME*, Lott GK*, Dutta A, Osborne JE, Reiser MB, Jayaraman V, "Two-photon calcium imaging from motion-sensitive neurons in head-fixed *Drosophila* during optomotor walking behavior" Nature Methods, 2010

Contents:

Treadmill Assembly Guide

Gus K Lott III, PhD

May 18, 2010

HHMI Janelia Farm Research Campus

19700 Helix Dr.

Ashburn, VA 20147

Documentation Package Contents

- **Treadmill_AssemblyGuide.pdf** – Description of the assembly process (this document)
- **Treadmill_SDK.pdf** - Description of the software interface to the treadmill system via the FTDI D2xx C API or as a virtual COM port serial device
- **TreadmillDemo.exe** - simple windows console application that connects to a connected treadmill device and streams/displays motion data. This application can be used to test that your system is functional.
- **Directory: Data Sheets**
    - Contains data sheets for the FTDI USB to Serial converter (and the D2xx programming guide), the Atmel ATMega644p microcontroller, and the Avago ADNS-6090 Optical Flow chip.
- **Directory: Electrical Design Files**
    - **Electrical_BOM.pdf** - The parts list for the electrical subsystems and identifiers to silkscreen and schematic labels for circuit elements.
    - **Schematic-Camera.pdf** - Circuit schematic for a single optical flow camera (the ADNS-6090 circuit)
    - **Schematic-MCUBoard.pdf** - Circuit schematic for the MCU board that connects to the PC and the two cameras.
    - **PCB-Camera.pdf** - To-Scale PCB design for camera
    - **Directory: Eagle Schematics**
        - Cadsoft Eagle schematic files for the MCU and camera system circuits (editable). The PDF schematics are derived from these files. They do not have associated PCBs
    - **Directory: ExpressPCB Files**
        - Contains the raw Express PCB files that may be used to order via www.expresspcb.com
    - **Directory: Gerber Files**
        - Contains sets of standard Gerber files used to order the system PCB's from any PCB vendor
- **Directory: Mechanical Design Files**
    - **Directory: Calibration Cube**
        - Detailed design blueprints for fabrication of a 6mm calibration target cube
    - **Directory: Camera Housing**
        - Detailed design blueprints for camera housing fabrication
    - **Directory: Inventor Files**
        - Autodesk Inventor Files for the Camera Housing (for advanced user modification)
- **Directory: Software**
    - **Directory: Firmware**
        - **Directory: Treadmill6090v2 May 2010**
            - A CodevisionAVR Project containing the C-code compiled into .hex binary file
        - **Treadmill6090v2.hex**
            - The compiled firmware binary for upload directly to the ATMega644p
    - **Directory: Matlab Code**
        - **Directory Treadmill IMAQ 2010a**
            - Visual Studio 2005 Project for Matlab Treadmill IMAQ adaptor
        - **motiondisplay.m –** Virtual COM Port Demo Code for motion tracking streaming
        - **motiondisplay_imaq.m –** IMAQ interface version
        - **treadmill.dll –** Compiled Matlab IMAQ Adaptor
        - **treadmillVideo.m –** Virtual COM Port Demo Code for camera preview
        - **treadmillvideo_imaq.m –** IMAQ interface version
    - **Directory: Treadmill Demo**
        - Visual Studio project for demo stand-alone C testing program

Assembly Overview

The assembly of a treadmill system consists of 3 steps

1. **Select & Order Lenses, Order PCBs, Order Parts, Order Mechanical Components**
2. **Populate PCBs, Assemble Camera Cables, Assemble Cameras**
3. **Upload Firmware, Test System via Software**

Complete Hardware Parts Check List

The file “Electrical_BOM.pdf” is provided with the documentation kit and contains links to all electrical components needed for a single treadmill system.

Part name	QTY	Part Number
Contents of Electrical_BOM.pdf		
Atmel AVR ISP mkII – In System Programmer	1	Digi-Key: ATAVRISP2-ND
MCU PCB	1	Included ExpressPCB or Gerber Files
Camera PCBs	2	Included ExpressPCB or Gerber Files
Camera Housings – Mechanical Construct	2	Included Blueprints (don’t forget screws)
Lenses	2	(see “Selecting Lenses” section below)
USB Cable – Type A to Type B	1	Digi-Key: AE9933-ND

Step 1: Selecting Lenses, Order PCBs, Order Mechanical Components

1.1 Selecting Lenses

The main innovation in this system compared to previous systems is the integration of arbitrary optics onto the flow meter. This allows for high speed, deterministic flow tracking applications on arbitrary moving objects from an arbitrary distance.

Figure: View into the Camera Housing through C-Mount Thread. 30x30 pixel array at the center

The 30x30 pixel array takes up a .05x.05” (1.27x1.27mm) square area nominally at the center of the c-mount thread in the camera housing and back 0.69” (17.53mm) from the outside face of the housing (lens flange focal distance). The diagonal of the square pixel grid is thus 0.07” (1.8mm).

To inscribe the pixel array within a circle of radius 0.035”, the pinhole camera equation provides an easy way to calculate the focal length of a required lens. It is, however, generally better to **greatly exceed the edge of the pixel grid** to remove the effect of any lens warping of the projected image.

A fixed focal length lens will make it easy to match the magnification factors between your two cameras (if you're using two), but with a calibrated target (see calibration cube section) with fixed line spacing coupled to the Matlab camera viewing program (treadmillVideo.m), you could match variable focal length lenses to get equal scaling of the moving field of interest.

The working distance (Z) is defined by the needs of your apparatus and is arbitrary. The imaged surface area (radius X) should be relatively flat and **flow within the field should not exceed acceleration rates defined in the ADNS-6090 datasheet** (measured in units "counts per inch" that doesn't directly match to pixels, thus trial and error may be required). Also, **make sure your lens passes IR**. Several good lenses that we used are available from closed circuit security camera vendors that frequently use IR light and c-mount lenses.

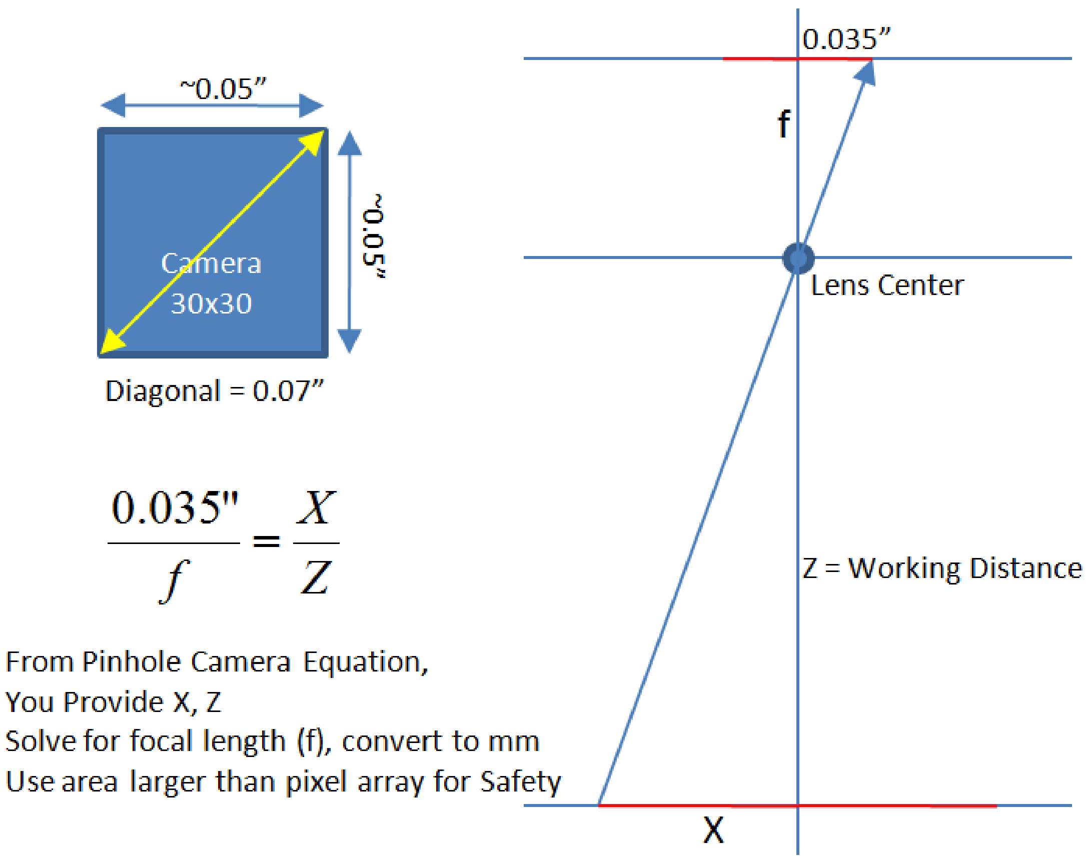


Figure: Pinhole camera equation for dimensions of the ADNS-6090 pixel array. Proportional triangles define the focal length of the lens used. One should greatly exceed the area of the array to reduce lens warping effects. Also make sure that the lens passes IR light.

1.2 Order Printed Circuit Boards & Parts

Printed Circuit Boards:

Relevant Files:
- Electrical Design Files\ExpressPCB Files***Final-CameraOnly.pcb***
- Electrical Design Files\ExpressPCB Files***Final-MCUonly.pcb***

or
- Electrical Design Files\Gerber Files\CameraOnly\
- Electrical Design Files\Gerber Files\MCUonly\

There are two paths to take to the creation of the required printed circuit boards. For a single system, you will need ONE of the MCU boards and TWO of the camera boards. Typically, PCB ordering costs don't change much with quantity (setup fees dominate), so order several of each.

The first path for ordering is to go directly through ExpressPCB.com. Their PCB editing software is free, comes with a schematic editor, and is quite easy to use. Their software is available for download at www.expresspcb.com and you can order the boards directly from a drop down menu in their software tool.

The second path to ordering involves the included Gerber file sets sent to an arbitrary PCB manufacturer. Gerber files are industry standard PCB milling definition files and are generally accepted by any board fabrication house. These files free you from your dependence on ExpressPCB.com for board fabrication, but the ordering process is a bit more complex. Each gerber directory contains a file for each circuit layer for each project. That's 8 layers per board plus a drill file defining where holes go. These 9 files per board will need to be provided to an alternative vendor for board fabrication.

Ordering Parts:

Relevant Files:
- Electrical Design Files\Electrical_BOM.pdf

We have included Digi-Key part numbers for all relevant components, but stocks vary, so you may have to find alternative parts or part vendors. In the case of parts with a new package layout, you may have to edit the ExpressPCB files directly to include the new package. The most difficult part to acquire seemed to be the ADNS-6090 chips. You may have more luck, but we got the chip when it first came out and had to buy them at about $5 each in a quantity of about 100 (via the ADNK-6090 kit). Sometimes, Avago will send sample parts and you may be able to get them to send you 2-5 6090 chips for your "prototype" if you ask nicely. We ordered ours from Avnet (link in the BOM file) with a considerable lead time, but you may have more luck with a little exploration.

Don't forget to include USB cables and an Atmel Programmer (parts listed above) in your order.

1.3 Order Mechanical Components or Fabricate In-House

Relevant Files:

- Mechanical Design Files\Camera Housing\
- Mechanical Design Files\Calibration Cube\

The mechanical components included in this documentation are limited to the camera housing itself with mounting holes. We also include Autodesk Inventor design files for the housing in case you wish to incorporate it into your own 3D design for your rig. We assume that you will have your own custom space requirements and will take the camera cases and mount them on your own rail system or otherwise to achieve whatever angular spacing you are allowed at whatever distance you need.

The included blueprints should allow any external machine shop to fabricate housings for your system. Anodized coating on the aluminum will reduce conductivity, reflections, and will prevent aluminum dust from potentially damaging the circuit or the optics.

We have also included design files for our 6mm calibration target cube which contained precision laser machined lines for centering and zoom-balancing both cameras in our fruit fly system.

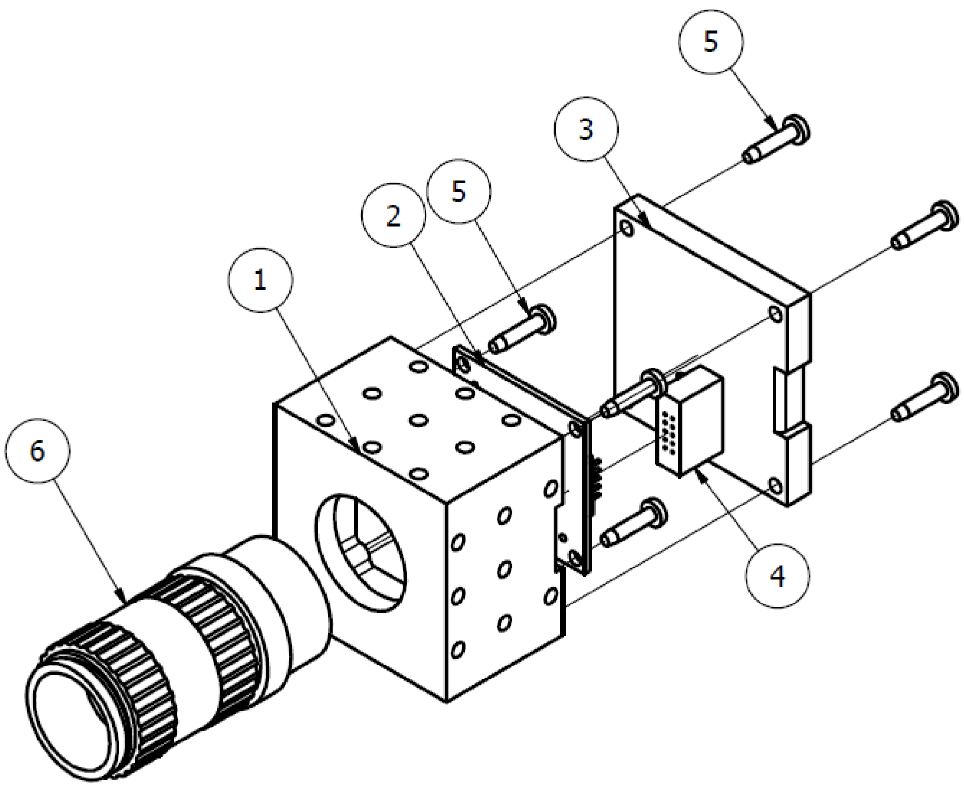


Figure: Camera Housing w/ Lens, Don't forget the 4-40 Screws!

Step 2: Populate PCBs, Assemble Cables, and Assemble Cameras

2.1 Populate MCU & Camera PCBs

Relevant Files:

- Electrical Design Files\Electrical_BOM.pdf
- Electrical Design Files\PCB-MCUBoard.pdf
- Electrical Design Files\PCB-Camera.pdf

MCU PCB

Labels in the bill of materials should match part labels on the PCB silk screen. An image of the assembled PCB (without the power indicator LED) is shown below. Note that the only components on the back side are the jumpers J8 & J9 which could drive included VCSEL laser diodes for illumination (see ADNK-6090 datasheet), but in the absence of the diodes, shorting the lines (with a jumper or otherwise) seems to be required for unimpeded system operation.

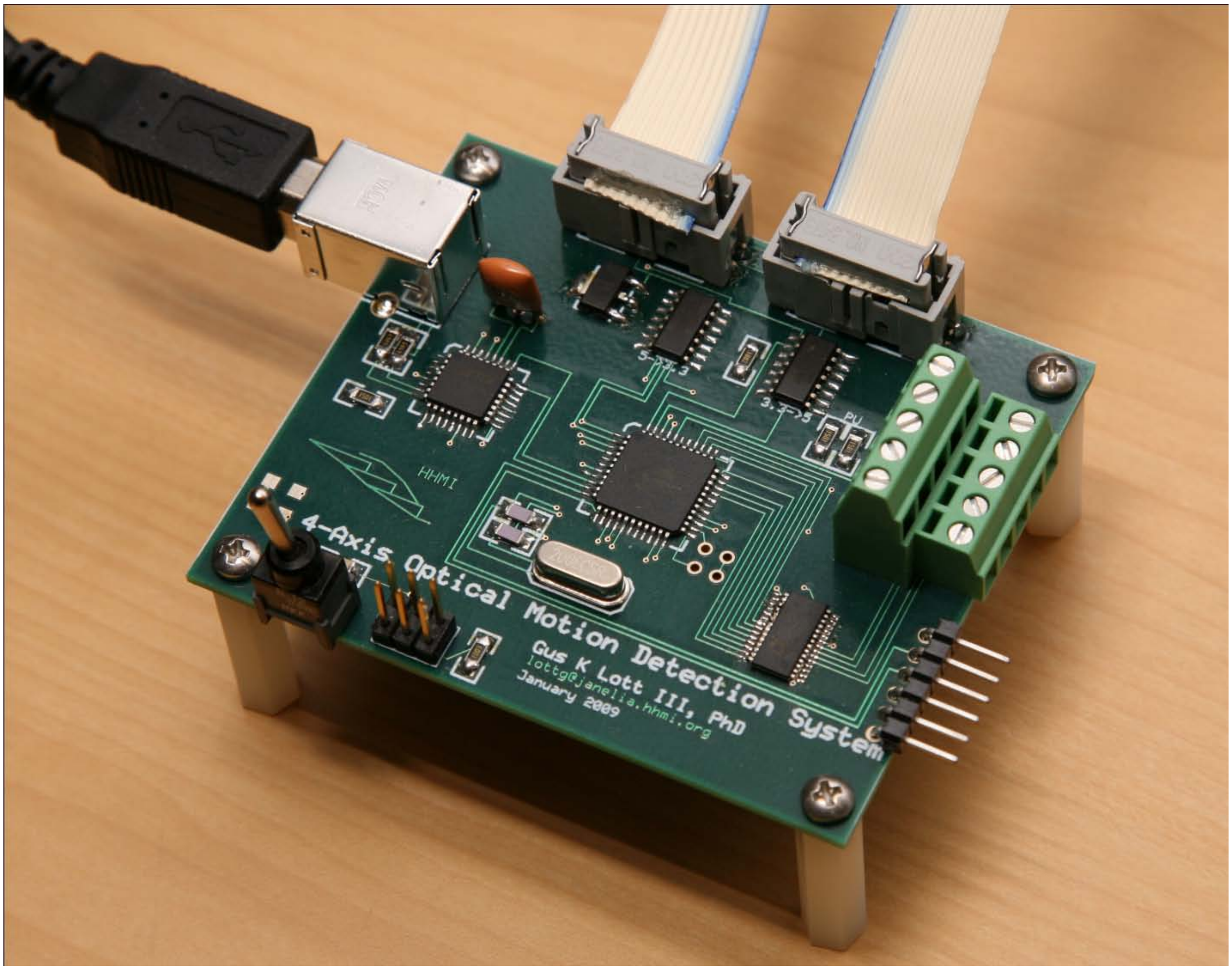


Figure: Fully Assembled Treadmill MCU board

Camera PCBs, Remove 6090 Bushing, protect Camera Array

When assembling the two camera PCBs, bill of material components match silk screen labels on the PCBs. There are no back side components. We recommend the first step in the assembly involve the placement of the ADNS-6090 camera chip as it is critical that the **chip body be flush with the PCB surface** to create a flat projection of the moving image on the 30x30 pixel array.

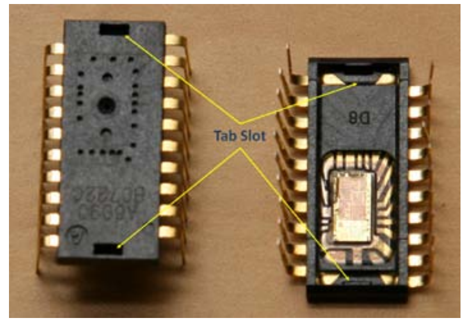


Figure: Removing the Plastic Bushing from the ADNS-6090 to expose the camera

The ADNS-6090 chip ships with a plastic bushing that directs light from their included optics (a lens designed for a surface 4mm away from the bottom of an optical mouse) onto the sensor. Fortunately, this bushing is only held in place by a pair of tabs, and a set of slots (see figure) allow you to remove the tab with a knife blade pushed through the slots. When the tab is removed, the chip will fit flush against the PCB to align the center of the sensor array with the camera thread (this is the optical axis of the system, so alignment is critical).

It is recommended that the assembly process begin with the application of a piece of tape to cover the hole through which the pixel array is exposed (on the back side of the PCB). This will prevent solder or flux from splashing onto the actual circuit elements in the camera chip.

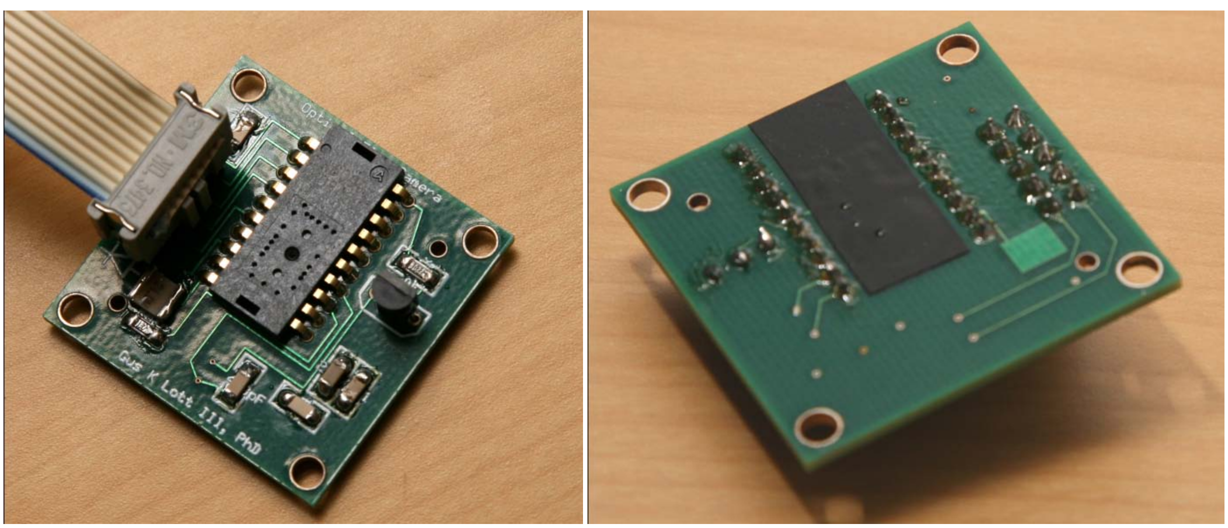


Figure: Fully Assembled Camera Board (You'll need 2), Note cable tab points in. Tape over the back of the camera can be removed later and protects the optical sensor during soldering.

2.3 Assemble Cables

The cable assembly consists of the parts described in the electronic parts list document used to populate the PCBs. You should be able to cut an arbitrary length of cable to meet your rig needs for camera placement, but at some point you may get signal degradation or cross-talk between data lines if the cable gets too long. In our work at Janelia, our cables were approximately 2 feet long. Make sure that the tab on the connector points out on both ends of the cable. A small press can be used to pierce the pins on the connector into the ribbon cable.

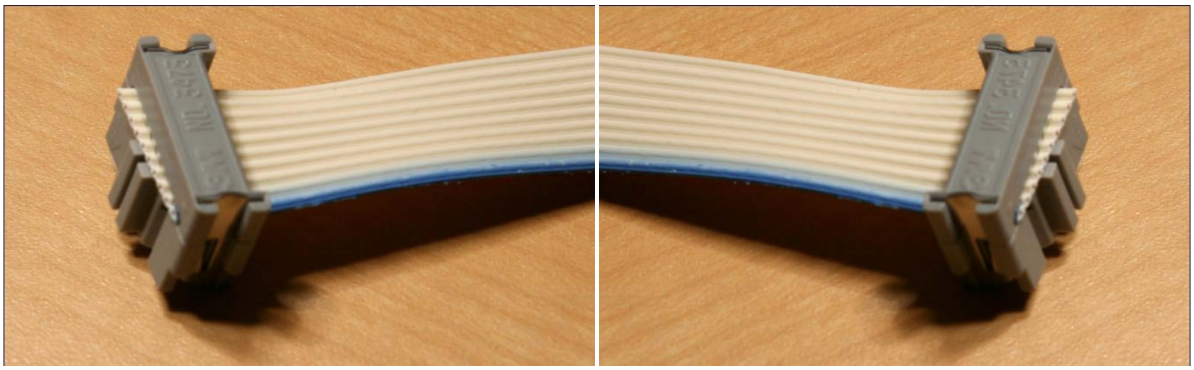


Figure: the camera cable connecting each camera to the MCU board.

2.4 Assemble Cameras

The camera assembly mainly consists of fitting the camera PCB to the internal dowel pins (see labels in figure). These pins assure that the image sensor is aligned at right angles to the camera housing and at the center of the c-mount lens thread.

Four internal 4-40 screws fix the PCB to the camera housing and four more screws mount the back plate to close up the camera housing. Don't forget to attach one end of the camera cable before closing the housing. The protective covering on the sensor can be removed through the c-mount thread or before you install the board in the camera case.

When viewing the camera from the back with the cable coming out of the left, the positive y-axis for motion detection is vertical and the positive x-axis is to the right.

Figure: alignment pins labeled for camera assembly

Step 3: Upload Firmware, Test System via Software

3.1 Uploading Firmware to the Treadmill System

Once the system is assembled and powered up, you will need to program the microcontroller with the treadmill application firmware.

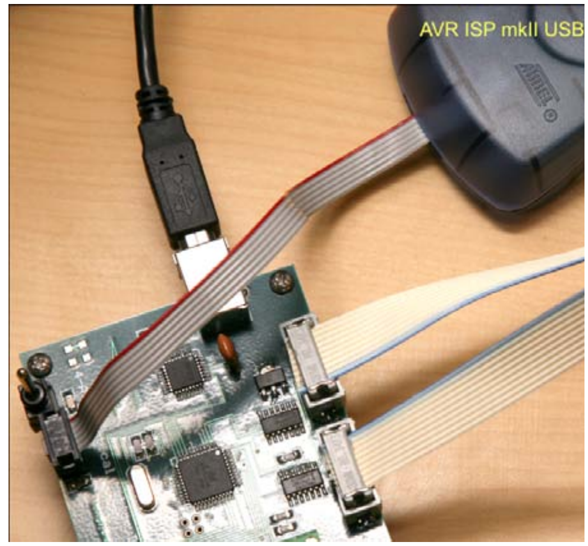


Figure: How to connect the AVR ISP mkII to the treadmill in system programmer pins.

Required Components:

1. Atmel AVR ISP mkII – In System Programmer
2. Atmel AVR Studio – Software (free download from Atmel.com)
3. Included in Documentation: *\Software\Firmware\Treadmill6090v2.hex*

The chip must be programmed as followed:

1. Connect your ISP to the treadmill system (see figure at right). A green light will appear on the ISP when plugged into the programming header correctly to a powered up treadmill system.
2. Open AVR Studio (we used version 4)
3. Close the “welcome to AVR studio” dialog
4. From the “Tools” menu, select “Program AVR” and “Connect”
5. From the “Platform list,” Select “AVRISP mkII” and hit connect (make sure your programmer is plugged into a USB port on your PC).
6. Under the “Main” tab, select “ATmega644p” from the Device drop down menu. Hitting the “read signature” button should return the chip signature and you should see a message “Signature matches selected device.” If this is not the case, make sure that you have the power switch in the on position on the treadmill and that the power indicator LED is active.
7. Under the “Fuses” tab, make the following changes to the default settings:
    a. Uncheck JTAGEN
    b. Uncheck CKDIV8
    c. Change SUT_CKSEL to “Ext. Crystal Osc. 8.0- MHz; Startup time: 258 CK + 65ms”
8. Hit Program (on the Fuses Tab)
9. Under the “Program” tab, within the “Flash” panel, select the location of the program “.hex” file included with the documentation (Treadmill6090v2.hex).
10. Hit the “Program” button in the “Flash” panel of the “Program” tab to upload the firmware.
11. Once complete, you should have an operational treadmill system.

3.2 Testing System via Stand-Alone app or Matlab Software

Relevant Files:
\TreadmillDemo.exe

We have included a stand-alone example windows console application for testing that your treadmill is operational without the requirement of a Matlab License. Make sure that you have installed the FTDI chip drivers:

FTDI Drivers for the Treadmill Interface:
http://ftdichip.com/Drivers/D2XX.htm
We used version 2.06.02 for both Windows XP and Windows 7.

Once installed, your device should appear as a virtual serial port in your system's device manager. Running the included TreadmillDemo.exe should produce output similar to that seen below. If the treadmill is not working, the program should time-out.

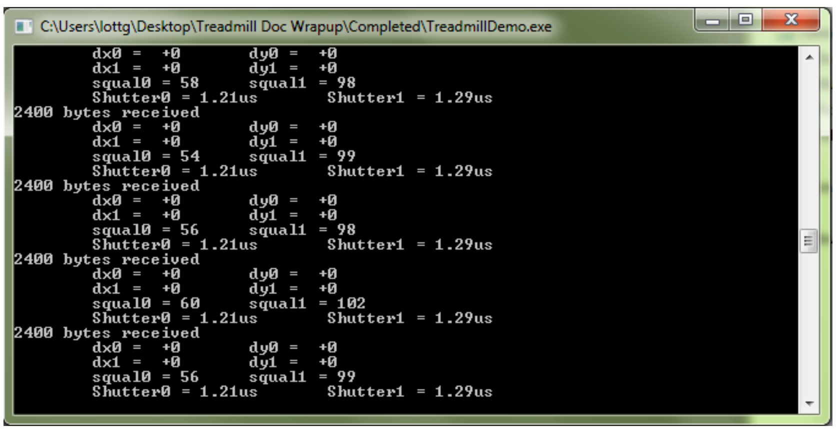


Figure: Example output of the included stand-alone windows console test program from a functional treadmill system.

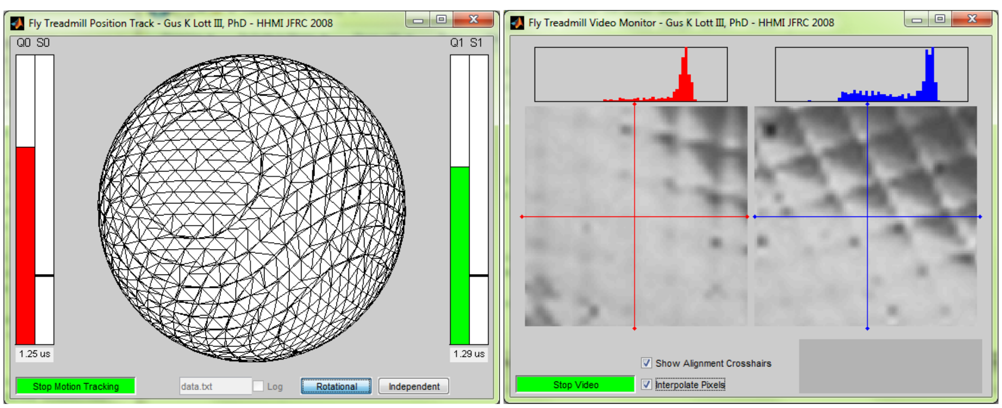
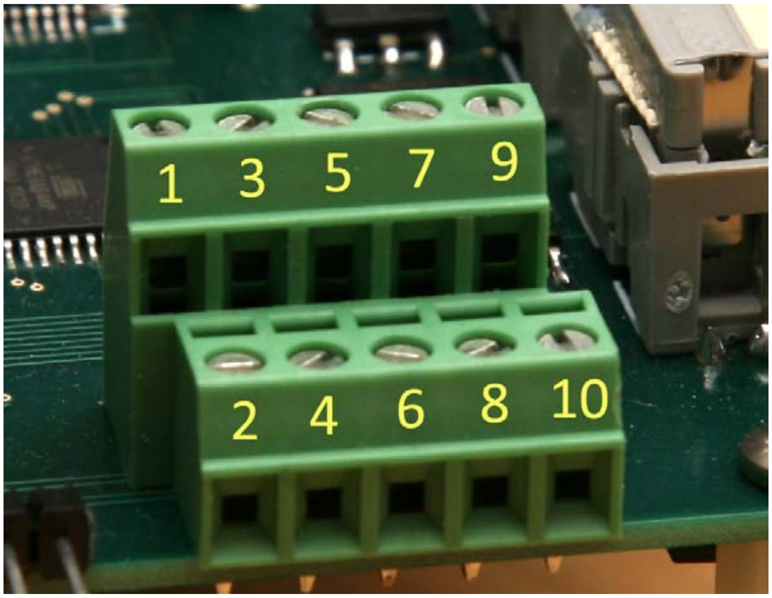


Figure: Example Matlab Applications to display motion and camera Views. See the included SDK documentation for more details

Wrapping Up: Concerns for your rig

The real-time interface (J7)
Included in the system is a 10 wire analog/digital real-time interface that generates clock signals for synchronization with other equipment (i.e. captured in parallel with electrophysiology) to determine high precision timing of motion data relative to other events in your apparatus.



Top View	
9	10
7	8
5	6
3	4
1	2

[1] **aCLK** – Analog Clock Signal: 5V digital line goes high whenever a new analog value is valid on each of the analog output lines (4,5,6,7)
[2] Reserved – digital I/O. Line 1&2 can operate as an I2C bus to control other devices
[3] **sCLK** – Sample Clock: 5V digital signal goes high when polling from camera (i.e. 4kHz)
[4] **Vx0** – X-Velocity from camera 0 – binned in 1ms intervals (4 samples) and centered around 2.5V with a fixed scaling per velocity tick. aCLK goes high when a new value is valid.
[5] **Vx1** – X-Velocity from camera 1
[6] **Vy0** – Y-Velocity from camera 0
[7] **Vy1** – Y-Velocity from camera 1
[8] **TRG** – Experiment Trigger: 5V Digital line goes high during entire sampling stream (when a start command is sent) and low when a stop command is sent.
[9] **5V** – USB Bus voltage for driving other peripherals, should draw less than 50mA
[10] **GND** – System Ground

Secondary Serial Interface (J6)
There is also a 6-Pin header socket next to the 10-pin interface that exposes a second serial bus from the controller. This bus can be converted to RS232 levels (to deterministically control experimental equipment), to USB via an FTDI Cable (i.e. TTL-232R) for connection to a PC, or can be fed directly into a UART port on another embedded system for transmit and receive. There is also a ground pin and two general purpose digital I/O that could be used for handshaking or any sort of 5V device triggering in addition to the real time interface.