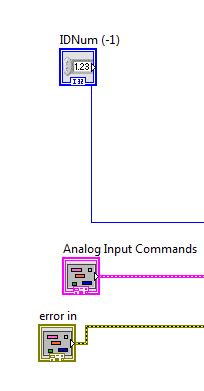
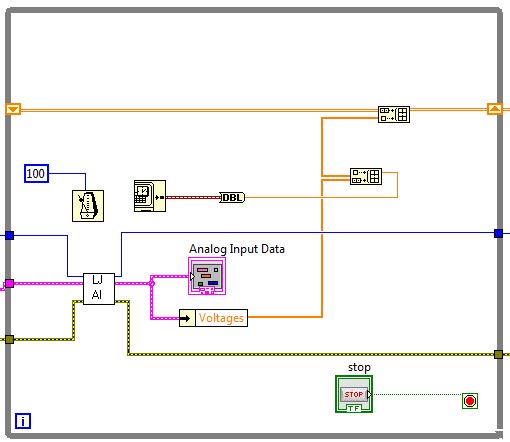
**1. Introduction**

In this lab, we need to use a real sensor “Optical Speed and Distance Sensor” to get data and process it in MATLAB. In the end, we get the offset, the transfer factor and the deviations of them.

**2. Extension of sensor module**



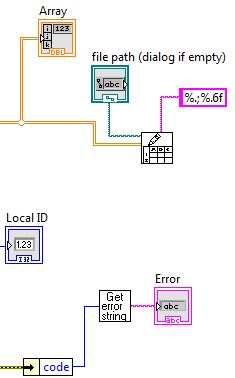
In last lab, we read from a \*.txt file, but now we connect to a real sensor and have an analog input.



We use loop to read data continually and the stop button is to end it. “labjackAD.vi” is provided for us. Then use the “Unbundle by Name VI” to access the value “Voltages” from the output cluster.

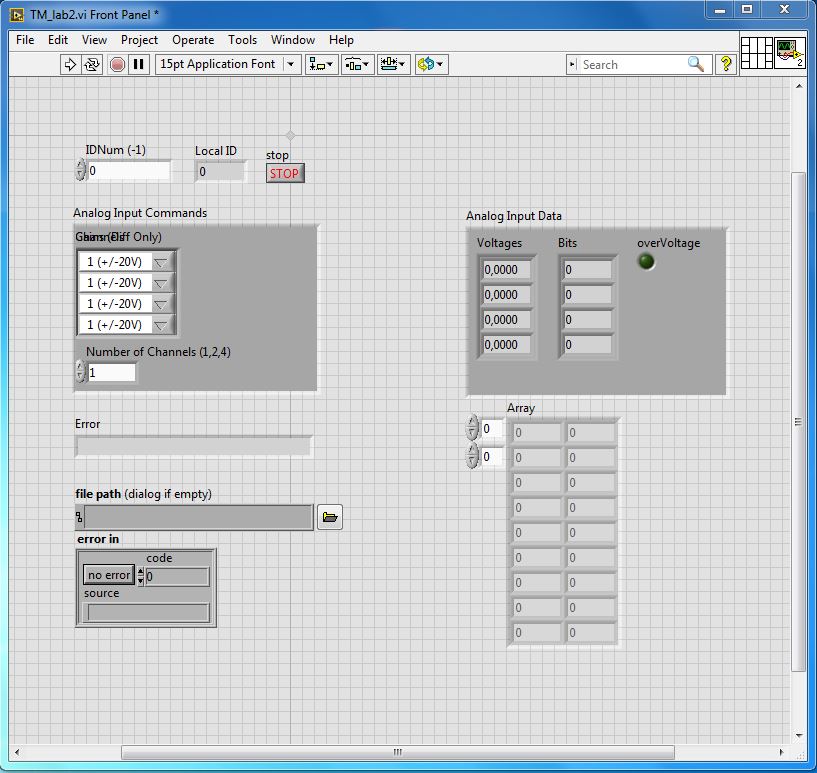
Also to reach an equidistant sampling rate, the timer is reserved. “Get Date/Time In Seconds” and “To Double Precision Float” are used for timestamp.

We build an array to store our output which are the timestamp and velocity respectively.



For output, considering we still need to process data in matlab, we convert the comma “,” in german version to “.” And write it to a file.

The front panel is



**3. Sensor calibration**

We hold our sensor still for 30 seconds to get the offset and decide to choose the first 100 records as our calibration data. With calculation in MATLAB, the result is

The standard deviation of offset is

The corresponding transfer factor k is derived from

We measured 10 times in a reference distances of 12.7 m, and the results are shown as follows.

k1 = 2.7138

k2 = 2.7465

k3 = 2.6831

k4 = 2.6911

k5 = 2.6555

k6 = 2.6911

k7 = 2.6927

k8 =2.6794

k9 = 2.6988

k10 = 2.6885

The mean and standard deviation of k are

In conclusion, we can get:

|  |  |
| --- | --- |
| offset | 0.0391 |
| standard deviation of offset C:\Users\Yihui\AppData\Local\Temp\ksohtml6832\wps3.jpg | 0.00532 |
| transfer factor *k* | 2.6941 |
| standard deviation of transfer factor C:\Users\Yihui\AppData\Local\Temp\ksohtml6832\wps6.jpg | 0.0224 |

So the transfer function is