

# Talk2Tobii

An Interface between the Tobii EyeTracker  
and MATLAB

Program User's Guide

Release: November 2007

Written by  
Fani Deligianni, PhD

Centre of Brain and Cognitive Development  
Birkbeck University,  
University of London

2007

## **Licence Agreement**

This software is provided 'as is' and without any implied support or guarantee, without even the implied warranty of merchantability or fitness for a particular purpose. UNDER NO CIRCUMSTANCES SHALL THE AUTHORS BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS PROGRAM. By downloading it you agree to use it at your own risk and that the author shall not be held accountable for any direct, indirect and incidental losses or damages, which you may incur.

## **Acknowledgments**

This work is supported from James S. McDonnell Foundation. The grant aims to 'Combine methods for assessing cognitive development in human infants and to promote the field of developmental cognitive neuroscience'.

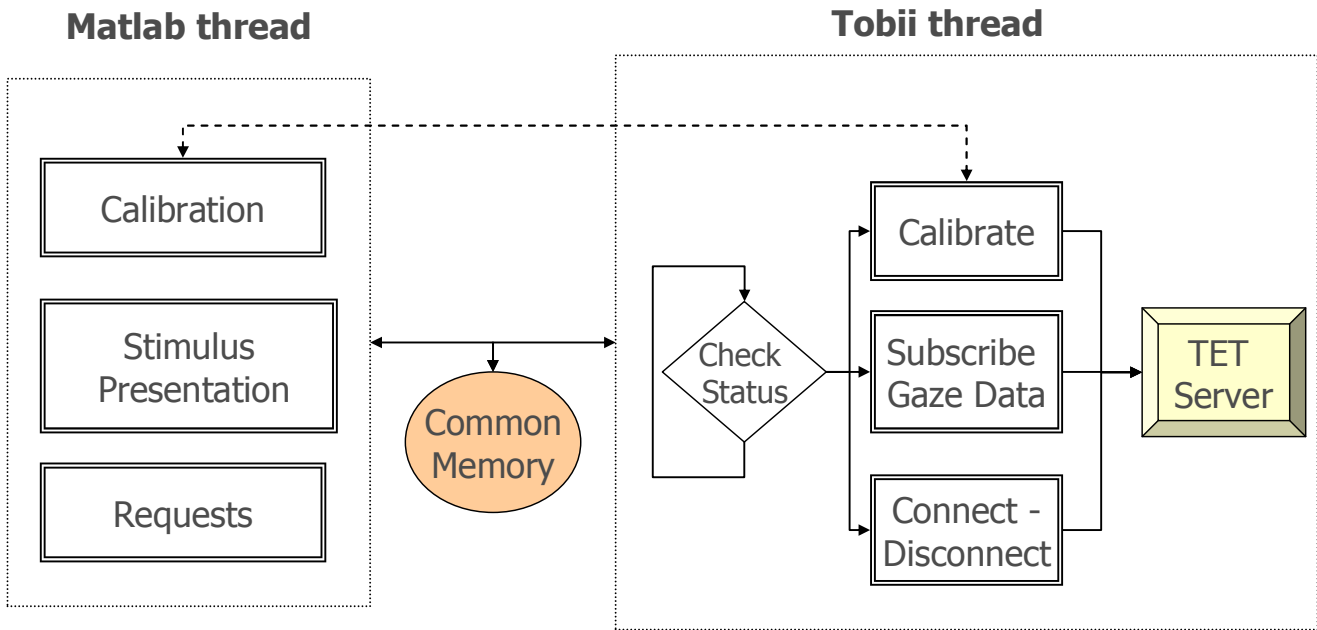
We thank Andrew T. Duchowski who has ported the Tet library in Linux and he kindly compiled it for intel-mac as well. His book 'Eye Tracking Methodology: Theory and Practice' was a valuable resource.

We are also thankful to Tobii technology that gave us permission to access and modify the source code for Tet library. Note that Tobii technology does not officially support the talk2tobii toolbox.

# Contents

Licence Agreement .....	2
Acknowledgments.....	3
Contents .....	4
Overview and Software Design .....	5
System Requirements and Computer's Configuration .....	6
Overview of functions.....	7
Connect - Disconnect.....	7
<i>talk2tobii ('CONNECT',hostname,port)</i> .....	7
<i>talk2tobii ('DISCONNECT')</i> .....	7
Calibration.....	8
<i>talk2tobii ('START_CALIBRATION', pnts, load_calibration, filename)</i> .....	8
<i>talk2tobii ('ADD_CALIBRATION_POINT')</i> .....	8
<i>talk2tobii ('DREW_POINT')</i> .....	8
<i>quality = talk2tobii('CALIBRATION_ANALYSIS')</i> .....	8
Start Tracking - Stop Tracking .....	9
<i>talk2tobii ('START_TRACKING');</i> .....	9
<i>talk2tobii ('STOP_TRACKING');</i> .....	9
Get Status - Clear History .....	9
<i>[status,history] = talk2tobii ('GET_STATUS')</i> .....	9
<i>talk2tobii ('CLEAR_HISTORY');</i> .....	10
Get Sample.....	10
<i>gazeData= talk2tobii ('GET_SAMPLE');</i> .....	10
Send Events.....	11
<i>talk2tobii ('EVENT',Event_Name, time, duration, 'nameOfField', value, ...);</i> .....	11
Record Eye Tracking data to memory .....	11
<i>talk2tobii ('RECORD');</i> .....	11
<i>talk2tobii ('STOP_RECORD');</i> .....	11
Save data to hard drive.....	11
<i>talk2tobii ('SAVE_DATA', eye_trackin_data, events, 'APPENDorTRUNK');</i> .....	11
<i>talk2tobii ('CLEAR_DATA');</i> .....	12

## Overview and Software Design

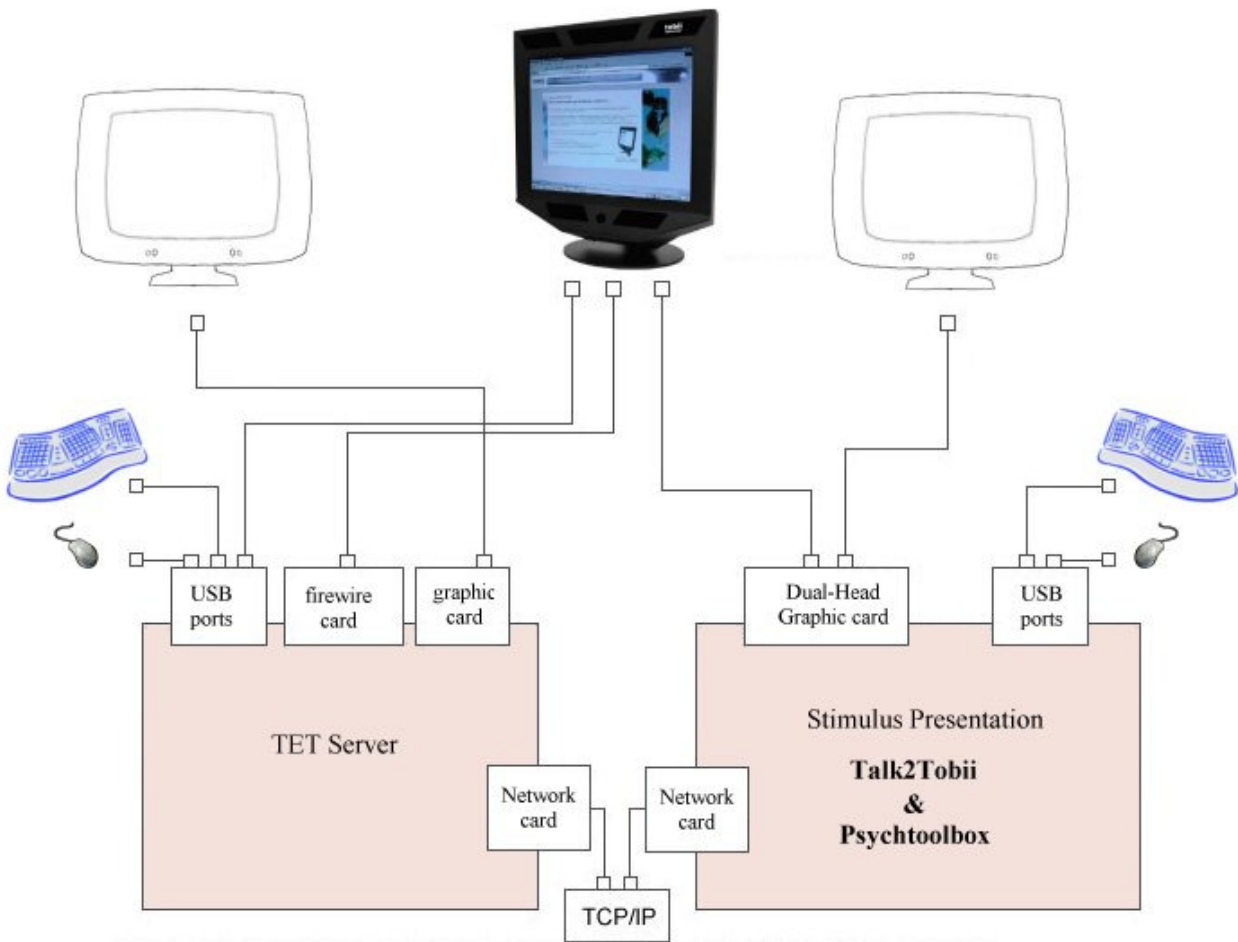


**Figure 1: Software Architecture**

The talk2tobii toolbox is an interface between the Tobii eye-tracker and Matlab. This software along with psychtoolbox can be used to create contingent eye-tracking applications and facilitate multi-modal acquisition of EEG, NIRS and eye-tracking data. It is based on multi-threaded technology to allow real-time eye-tracking and interactive stimulus presentation and it is implemented as a mex function in C++. Talk2tobii is largely inspired by the EyeLink Toolbox for Matlab and it provides similar functionality for the Tobii eyetracker.

Figure 1 presents a simplified version of the overall architecture of the software. Users should be aware that two threads, Matlab and Tobii, are running in parallel and not necessarily in synchrony. In general, users interact directly with the Matlab thread, which they use to prepare their data and present the stimulus. The Tobii thread is initialised through the Matlab thread and it implements the communication between the application and the Tobii Eye Tracker (TET) server. The communication is based on TCP/UDP protocol and, thus, in theory talk2tobii may run on the same or different machine as the TET server. The two threads share common memory variables that describe the current status of the eye-tracker (connected, eye-gaze subscription, calibrating and so on) and request data or change of the eye-tracking state (connect- disconnect, start tracking, start calibration, load the latest gaze data into matlab and so on). The two threads interrupt each other only during calibration, since they need to be in synchrony in order to acquire data for each point successively while a stimulus is presented to attract attention at that point.

## System Requirements and Computer's Configuration



**Figure 2: Computer's Configuration**

This release of talk2tobii runs only on mac-intel OSX machines and it requires the installation of MATLAB and Psychtoolbox. For this reason, two computers are required to acquire eye-tracking contingent data, see Figure 2. One is the standard Windows PC (left) that runs the TET server and is connected to the eye tracker via the USB port and the firewire card to acquire the eye tracking data. The second one is a mac-intel computer that presents the stimulus (right). The latter is connected to the stimulus presentation screen via an RGB cable only. The two computers need to be connected through their network cards. Make sure that the firewall on the PC is off.

The software has been compiled with Xcode 2.4.1 on MacBook Pro OS 10.4.10 and it has been tested with Tobii 1750, Matlab R2007a and Psychtoolbox 3.0.8 flavor beta, which corresponds to SVN Revision 703. Normally, it should work on any intel-mac that runs OSX and matlab (version 6 or later) along with Psychtoolbox version 3 or later. It should also be able to work with the portable eye-tracker x50 as well as Tobii 1750.

## Overview of functions

The Talk2Tobii toolbox offers the functions summarised in table 1. They are classified according to their function to the following categories: Connect-Disconnect, Calibration, Start Tracking – Stop Tracking, Log record – Error handling, Receive on-line data, Store Eye Tracking data and events to memory and to hard-drive.

**Table 1: Brief Description of the Commands available in the Talk2Tobii Toolbox**

Command	Function
CONNECT	Request to initialise communication with TET server
DISCONNECT	Request to close communication with TET server
START_TRACKING	Request to start subscribing gaze data
STOP_TRACKING	Request to stop gaze data' subscription
GET_STATUS	It returns an array of variables that describe the status of the Tobii thread
CLEAR_HISTORY	Clear history, which is a vector tracking the status of Tobii thread
GET_SAMPLE	Receive on-line data that can be processed in Matlab
START_CALIBRATION	Start Calibration
ADD_CALIBRATION_POINT'	Informs the Tobii thread that the drawing of the next point has started and it blocks the Tobii thread till the drawing finish
DREW_POINT	It informs the Tobii thread that the drawing of the calibration point has finished to unblock the thread
CALIBRATION_ANALYSIS	It returns data that can be analysed to investigate the quality of the calibration
EVENT	Send events and store them to the memory
RECORD	Store gaze data to memory
STOP_RECORD	Stop storing gaze data to memory
SAVE_DATA	Save stored gaze data and events to a file
CLEAR_DATA	Discard gaze data and events from memory

### **Connect - Disconnect**

*talk2tobii ('CONNECT',hostname,port)*

It sets a flag that results in changing the execution of the tobii thread to connect to the TET server via TCP/IP. 'hostname' is the IP address of the PC that runs the TET server. 'port' is the port that the TET\_server is connected. For the 1750 and the x50 tobii eyetracker this port is 4455. This function does not return any value. If an error has occurred, it cannot be detected with this function. Use 'GET\_STATUS' to check the status of the connection with the TET server and to detect any errors.

*talk2tobii ('DISCONNECT')*

It sets a flag that results in changing the execution of the tobii thread to disconnect from the TET server. If the tobii thread is not connected with the TET server, nothing happens. This function does not destroy the tobii thread. Tobii tread is destroyed when the talk2tobii mex is cleared or exit matlab.

## Calibration

Three commands are combined to calibrate the Tobii eye tracker: 'START\_CALIBRATION', 'ADD\_CALIBRATION\_POINT' and 'DREW\_POINT'. If there are not used properly the tobii thread MAY LOCK and do not allow further interaction. See example code of how to use them.

*talk2tobii ('START\_CALIBRATION', pnts, load\_calibration, filename)*

This function implements the following steps:

1. It defines an m-by-2 array (pnts) that contains the coordinates of the calibration points. Where m is the number of points and the two columns correspond to the x and y coordinates respectively. The coordinates are normalized from 0 to 1.
2. load\_calibration sets a flag that defines if a stored calibration will be loaded (This feature is not fully implemented yet so flag should be set to 0). 'filename' is the filename of the calibration file to be loaded if load\_calibration is 1 or the filename to store a successful calibration if load\_calibration is 0.
3. Sets a flag that allows the tobii thread to start calibration

*talk2tobii ('ADD\_CALIBRATION\_POINT')*

It informs the tobii thread that the drawing of the next point has been started and it blocks the thread till the eye tracker is ready to continue.

*talk2tobii ('DREW\_POINT')*

It signals the tobii thread that the drawing of the calibration point has been finished and calibration can be continued with the next point.

*quality = talk2tobii('CALIBRATION\_ANALYSIS')*

It returns an array of the data acquired during calibration and their accuracy. It provides the target points, the resulting mapped points and an indication if a point was discarded. See Table 2 for more details.

**Table 2: Fields contained in the 'quality' of calibration**

Field	Description
truePointX	X coordinate for point where it was displayed for the user
truePointY	Y coordinate for point where it was displayed for the user
leftMapX	Left eye, X coordinate for mapped point
leftMapY	Left eye, Y coordinate for mapped point
leftValidity	Left eye: (-1) - was not found, (0) - found but not used, (1) – used
rightMapX	Right eye, X coordinate for mapped point
rightMapY	Right eye, Y coordinate for mapped point
rightValidity	Right eye: (-1) - was not found, (0) - found but not used, (1) - used.

## Start Tracking - Stop Tracking

*talk2tobii ('START\_TRACKING');*

It sets a flag that allows the tobii thread to start subscribing gaze data. If the tobii thread is not connected with the TET server nothing happens. This function does not return any value. If an error has occurred , it cannot be detected with this function. Use 'GET\_STATUS' to check the status of the connection with the TET server and to detect any errors.

*talk2tobii ('STOP\_TRACKING');*

Sets a flag that allows the tobii thread to stop the subscription of gaze data. If the tobii thread is not connected with the TETserver nothing happens. This function does not return any value. If an error has been occurred cannot be detected with this function. Use 'GET\_STATUS' to check the status of the connection with the TET server and to detect any errors.

## Get Status - Clear History

*[status,history] = talk2tobii ('GET\_STATUS')*

It is used mainly for debugging to allow insight in the status of Tobii thread and the errors that have been occurred. 'status' is an array with 0 or 1 describing bits that correspond to the description at Table 3, respectively. 'history' is an m-by-2 array that records all the main calls to the TET API and whether they were successful. It also includes a record of the timestamp as it had been reported by GetSecs after the function's call. The first column of this array contains integer values from 1-12 if an error has occurred or integer values above 100 if an error has not occurred. See Table 4 for more details.

**Table 3: Listing and Description of the Fields included in 'status'**

Status	Description
TET_API_CONNECT	It has been requested to connect to the TET server
TET_API_CONNECTED	The communication with the TET server has been initialized successfully
TET_API_DISCONNECT	It has been requested to terminate the connection with the TET server
TET_API_CALIBRATING	It has been requested to start calibration
TET_API_CALIBSTARTED	Previous calibration has been cleared successfully and the calibration process has been started.
TET_API_RUNNING	It has been requested the subscription of eye tracking data
TET_API_RUNSTARTED	The subscription of gaze data has been initialized successfully
TET_API_STOP	It has been requested to stop the subscription of gaze data
TET_API_FINISHED	The Tobii thread has exited
TET_API_SYNCHRONISE	It has been requested to synchronise the clocks
TET_API_CALIBEND	Calibration has finished
TET_API_SYNCHRONISED	The clocks at the host and remote computer has been synchronised

**Table 4: Listing and Description of ‘history’**

Flag	Description
0	Problem initializing tobii (Tet_Init has failed)
1	Problem connecting with tobii (Tet_Connect failed)
2	Problem clearing calibration (Tet_CalibClear failed)
3	Problem adding Calibration point (Tet_CallibAddPoint failed)
4	Warning: Problem calculating and setting calibration (Tet_CalibCalculateAndSet failed)
5	Warning: Problem getting calibration results (Tet_CalibGetResult failed)
6	Warning: Problem saving calibration' (Tet_CalibSaveToFile failed)
7	Warning: Synchronisation failed' (Tet_Synchronise failed)
8	Problem starting tracking! EyeTracker will disconnect (Tet_Start failed)
9	Warning: Problem loading calibration file (Tet_CalibLoadFromFile failed)
100	Connecting with tobii...success (Tet_Connect was successful)
200	Clearing calibration...success (Tet_CalibClear was successful)
300	Adding calibration point...success (Tet_CallibAddPoint was successful)
400	Calculating and setting calibration...success (Tet_CalibCalculateAndSet was successful)
500	Calibration results have been obtained (Tet_CalibGetResult was successful)
600	Calibration have been saved (Tet_CalibSaveToFile was successful)
700	Synchronized maximal error... (Tet_Synchronise was successful)
800	Starting track...success (Tet_Start was successful)
900	Calibration has been loaded (Tet_CalibLoadFromFile was successful)

*talk2tobii ('CLEAR\_HISTORY');*

Discard previous history records. See the 'GET\_STATUS' for more information on what history contains.

## **Get Sample**

*gazeData= talk2tobii ('GET\_SAMPLE');*

Use this function to receive online gaze data. It returns an array 'gazeData' with the order and fields described in Table 5.

**Table 5: Listing and Description of ‘gazeData’**

Status	
x coordinate of the left eye	
y coordinate of the left eye	
x coordinate of the right eye	
y coordinate of the right eye	
time in Sec returned from the TETserver	
time in mSec returned from the TETserver	
left eye validity	Validity indicates how likely is it that the eye is found
right eye validity	Validity indicates how likely is it that the eye is found
	0 - Certainly (>99%)
	1 - Probably (80%)
	2 - (50%)
	3 - Likely not (20%)

	4 - Certainly not (0%)
left eye position - x coordinate	Eye position seen on the camera plane
left eye position - y coordinate	Eye position seen on the camera plane
right eye position - x coordinate	Eye position seen on the camera plane
right eye position - y coordinate	Eye position seen on the camera plane

---

## **Send Events**

*talk2tobii ('EVENT', Event\_Name, time, duration, 'nameOfField', value, ...);*

Use this function to send events from Matlab thread to the talk2tobii mex file. 'Event\_Name' is a string that describes the event. 'time' specifies the time the event started. This time should be in accordance with Psychtoolbox to allow synchronization with the gaze data. See talk2tobii('SAVE\_DATA') for more details. 'duration' specifies the time that the event last (set constant if it is not required). An unlimited number of pair values can be specified with the following format: 'nameOfField', 'value'. 'nameOfField' is a character string that describes the field. 'value' is numerical value that corresponds to this field.

## **Record Eye Tracking data to memory**

*talk2tobii ('RECORD');*

It calls the TALK2TOBII('SYNCHRONISE') and it sets a flag to start recording the eyetracking data. Data are stored in memory and they are not saved on hard drive unless 'SAVE\_DATA' is called. This function does not start the subscription of eye tracking data. Normally 'START\_TRACKING' is called prior to this function.

*talk2tobii ('STOP\_RECORD');*

Sets a flag that prevents further eye tracking data to store on memory

## **Save data to hard drive**

*talk2tobii ('SAVE\_DATA', eye\_tracking\_data, events, 'APPENDorTRUNK');*

It writes both the eye tracking data and the events to files in ASCII format. 'Eye\_tracking\_data' specifies the filename that will be used to store the data collected from tobii. The eye tracking data are stored in columns in the order presented in Table 6. 'events' specifies the filename that it will be used to store the events as they are specified during an 'EVENT' call. A '#START timestamp' provides a timestamp of when gaze data subscription started. This time is acquired with a call to the psychtoolbox function 'GetSecs'. This information is used to synchronize eye tracking data and events. 'APPENDorTRUNK' should be either 'APPEND' to allow appending data to existing file or 'TRUNK' to delete any previous data stored in the specified file.

**Table 6: Listing and Description**

Status	
1. time in sec	10. right camera eye position - y coordinate
2. time in msec	11. left eye validity
3. x gaze coordinate of the left eye	12. right eye validity
4. y gaze coordinate of the left eye	13. diameter of pupil of the left eye
5. x gaze coordinate of the right eye	14. diameter of pupil of the right eye
6. y gaze coordinate of the right eye	15. distance of the camera from the left eye
7. left eye position - x coordinate	16. distance of the camera from the right eye
8. left eye position - y coordinate	
9. right eye position - x coordinate	

*talk2tobii ('CLEAR\_DATA');*

Discard eye tracking data and events stored in memory