Optical Gesture & Proximity Sensing IC

Evaluation Board EVB75030V2 - Installation Manual

Melexis
INSPIRED ENGINEERING

Software Installation

I : mbed Microcontroller Firmware

EVB75030V2 has been shipped with dedicated firmware visible as
file “EVB75030V2_vx.x.x.bin” on the microcontroller.
This firmware allows communication between the MLX75030 and a pc via a terminal.
A detailed API is available and will be described on the following slides.
In order to communicate with the mbed via a Windows OS a driver has to be installed.
The install procedure is described in this URL and the driver can be downloaded here.
Note : On WindowsXP this driver has to be installed for each mbed separately.

II : National Instruments Labview Interface Layer

We have developed a “first order” interface, based on the API, that visualizes the sensor output, combines different registers settings and has an integrated simple left & right “swipe” detection algorithm. Robust proximity sensing is available on both channels independently.
In order to use this application you will need to install 3 separate parts :
1) NI Labview Runtime (to run any Labview executable)
2) NI Visa Runtime (to use any serial communication in Labview)
3) MLX75030 application program

Part 1 & Part 2 are combined in the following installation file or available via the official NI website (NI Labview Runtime & NI Visa Runtime) and Part 3 can be downloaded here.

EVB75030V2 API

The communication between the µC and the pc is serial.
To interface with a serial COM port you can use any terminal application.
(for example Putty, HyperTerminal, TeraTerm, …)

At startup the system automatically outputs data with a default framerate of 30 FPS.
In this continuous output mode it will accept input commands (described on the next slides) terminated with a carriage return character. (↵ ENTER)

Most commands exists out of a character (A, B, C, D, ….) followed by a number.
Every API command returns a string %s as confirmation.

The only way to turn the system OFF is by using the F0 (Framerate = 0) command.
In this state it will only be revived by A or M commands.

Important : The system has to run at a fixed framerate in order to maintain a constant duty cycle.
Alternating duty cycle(s) could affect LED performance.
However it's possible to change the duty cycle
based on different API commands.

Note : A fixed framerate also facilitates the development & optimization of software algorithms, like a "swipe" detection algorithm.

EVB75030V2 API Command List (1)

⇒ **A**%d : Enables temporal averaging on the microcontroller.
This function has a dedicated data output format that includes mean & STD.
%d = digital number in the range [2-99]

⇒ **D**%s%s%s%s%s : Enables HW DC Compensation.
The HW DC compensation consists out of 5 different parameters, kneepoints.
%s = hex. string in range [0-F]
D0 : HW DC compensation disabled
DFFFFF : HW DC compensation maximum

⇒ **F**%d : Changes the framerate, the maximum framerate is limited by serial baud rate.
%d = digital number in range [0-299]
F0 : System OFF
F1-299 : Changes framerate

⇒ **G**%d : Changes Data Output Format.
G0 : default
G1 : for use with MLX internal Python scripts
G2 : for use with “old” Labview interface (v1.2, v1.5)

EVB75030V2 API Command List (2)

⇒ **K%d** : Changes LED DAC value ChannelA = Register 0x1
%d : digital number in range [0-255]

⇒ **L%d** : Changes LED DAC value ChannelB = Register 0x3
%d : digital number in range [0-255]

The correlation between the DAC value and LED peak current is piece-wise linear.
A corresponding table can be found here.

⇒ **M%d** : Internally the system runs at a fixed framerate.
Command M1-99 outputs 1-99 measurements.
%d : digital number in range [0-99]
M0 : continuous output mode
M1-99 : output of 1-99 frames

⇒ **P%d** : Changes the amount of LED pulses per frame = Register 0x5
%d : digital number between (0-15, default value = 4)
A selection of 2,4,6,8,10,12,14,16,18,20,22,24,26,28,30 or 32 pulses can be made.
More information can be found p45 of Datasheet MLX75030 Rev005.

EVB75030V2 API Command List (3)

⇒ **R**%d : Read MLX75030 register(s).
%d : digital number in range [0-15]
R0-14 : Register 1-14
R15 : Read all registers simultaneously

⇒ **S**%d : Enable SW Compensation.
Not implemented at this moment

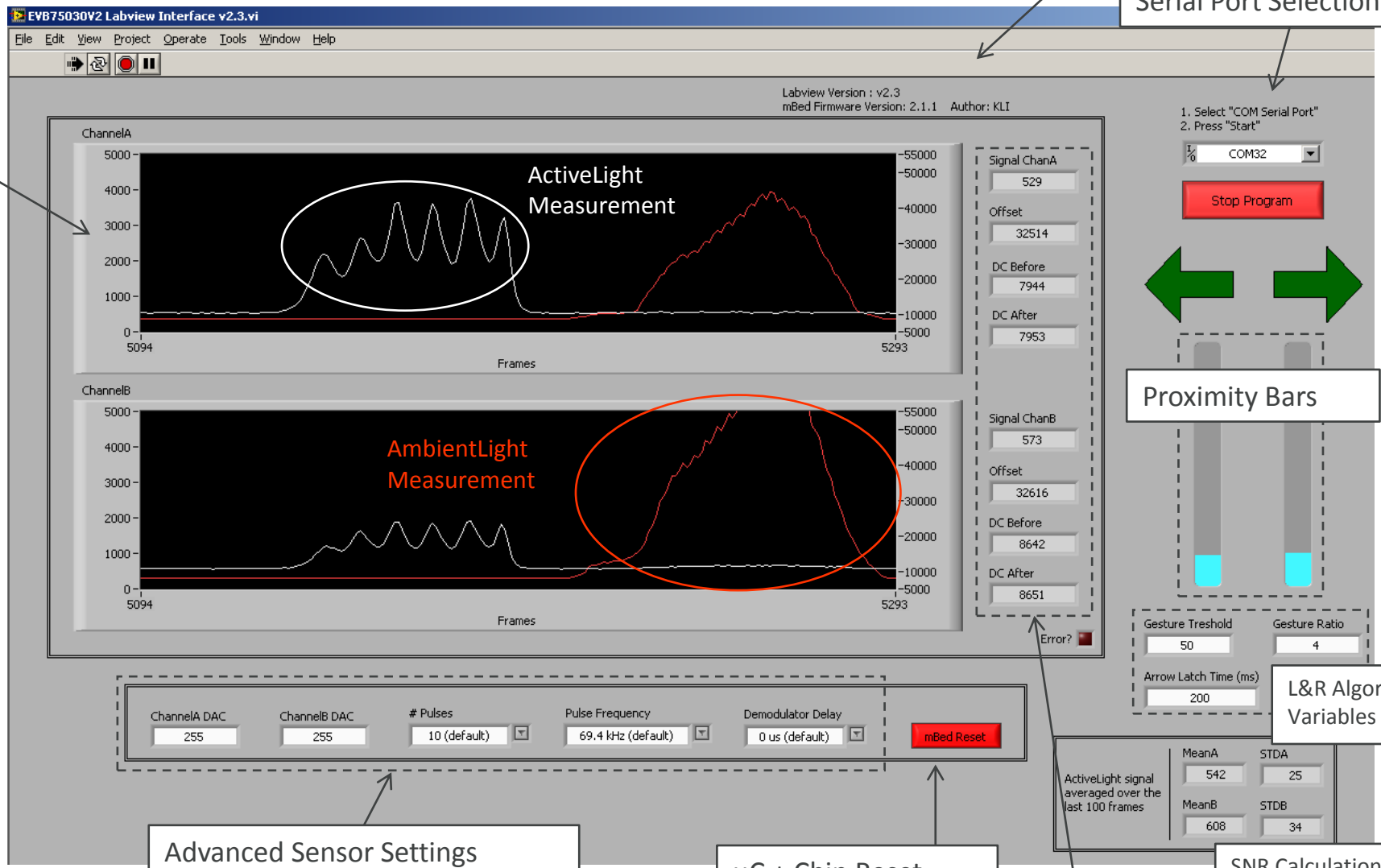
⇒ **T**%d : Changes MLX75030 internal demodulator delay. (= phase shift)
%d : digital number in range [0-15], default value is 0
More information can be found p42 of Datasheet MLX75030 Rev005.

⇒ **U**%d : Change pulse modulation frequency.
%d : digital number in range [0-7], default value is 4
More information can be found p46 of Datasheet MLX75030 Rev005.

⇒ **V**%d : Returns current firmware version of the microcontroller.

⇒ **X**%d : Initiates a microcontroller reset, all registers settings will change to default.

EVB75030V2 Labview Interface v2.3



Version Indicator(s)

Serial Port Selection

You can manually adjust Y-axis min. & max.

Proximity Bars

L&R Algorithm Variables

Advanced Sensor Settings

µC + Chip Reset

Raw Sensor Data

SNR Calculation of the last 100 frames

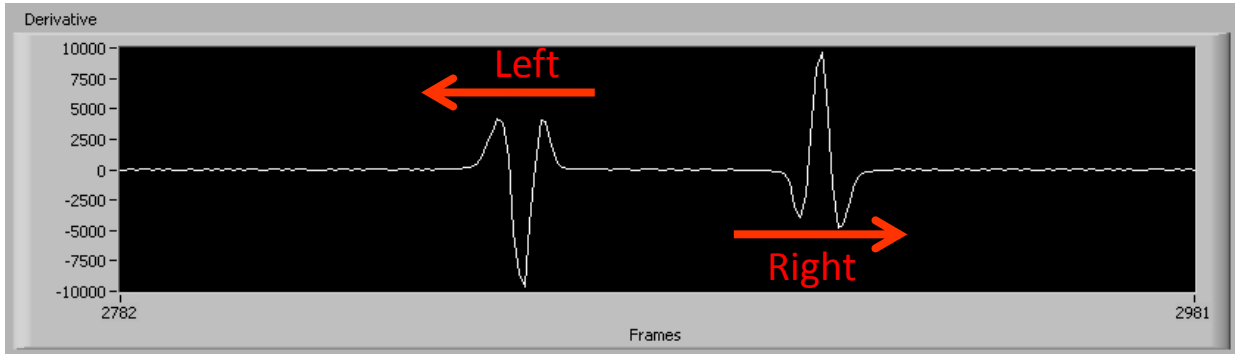
Melexis
INSPIRED ENGINEERING

EVB75030V2 Left & Right Gesture Recognition Algorithm

- A “derivative” value is calculated on each frame

$$Derivative(x) = \big(A(x) - B(x)\big) - \big(A(x-1) - B(x-1)\big)$$

- These values are stored in an *Array* which holds the last 6 frame values
- Gestures are recognized as a waveform :



- This waveform is split in 3 recognizable parts
- A gesture is only recognized if all 3 parts are *true*
- *“Left Gesture”* :

$$IF\ (max(Array[0:1]) > Treshold) \\ \&\ (min(Array[2:3]) < -Ratio \times Treshold) \\ \&\ (max(Array[4:5]) > Treshold)$$

- *“Right Gesture”* :

$$IF\ (min(Array[0:1]) < -Treshold) \\ \&\ (max(Array[2:3]) > Ratio \times Treshold) \\ \&\ (min(Array[4:5]) < -Treshold)$$