

Using the **ADUX1020** for Gesture Recognition

by Tyler Ray

INTRODUCTION

The **ADUX1020** is an integrated optical sensor and front-end device capable of advanced single point gesture recognition and proximity detection. The optical sensor enables sensitivity to the intensity and angle of incident light. Robust operation in a variety of noisy environments is made possible by a combination of optical and electronic ambient light filtering. The chip can be configured in multiple modes to enable touchless gesture recognition and control, or proximity detection.

GESTURE RECOGNITION BASICS

The sensitivity of the **ADUX1020** to the angle of incident light allows the chip to detect gestures using only the integrated optical sensor and a single external LED. The angle of incident light can be calculated from the four output channels on the chip when it is set to sample mode. The following procedure must be used for gesture recognition functionality:

1. Prior to the operation of the **ADUX1020**, calibrate the clocks. See the **ADUX1020** data sheet for information on how to calibrate the 32 kHz and 32 MHz clocks.
2. Set the **ADUX1020** mode of operation to sample mode by writing 0x3 to Register 0x45, Bits[3:0].
3. Collect the data measured by the device. See the **ADUX1020** data sheet for instructions on how to read data from registers using the first in, first out (FIFO) and interrupts. Data is available directly from data registers or from the 64-byte FIFO in Register 0x60, Bits[15:0].
4. The data in the four output channels of the **ADUX1020** calculates the angle of incident light. After the Channel X₁, Channel X₂, Channel Y₁, and Channel Y₂ data are collected, calculate the angles and intensity by the following equations:

$$\text{Horizontal angle: } \theta_x = \frac{X_1 - X_2}{X_1 + X_2} \text{ (radians)}$$

$$\text{Vertical angle: } \theta_y = \frac{Y_1 - Y_2}{Y_1 + Y_2} \text{ (radians)}$$

$$\text{Intensity: } I = X_1 + X_2 + Y_1 + Y_2 \text{ (codes)}$$

Examples of this data are shown in Figure 1 and Figure 2.

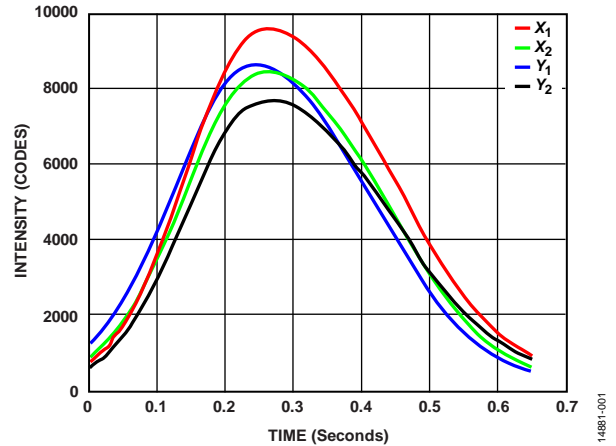


Figure 1. Intensity vs. Time, Raw Data Output for the **ADUX1020**

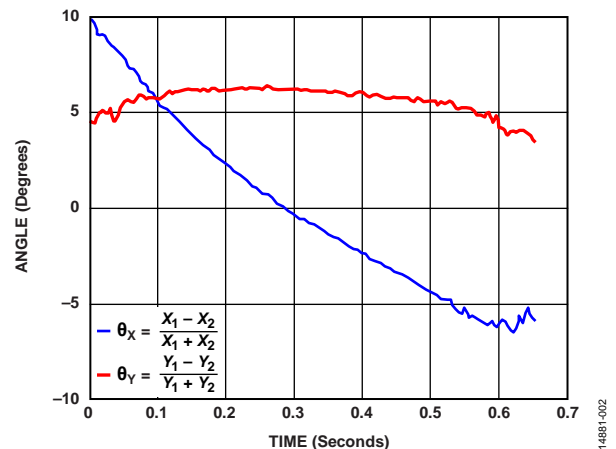


Figure 2. Angle vs. Time, Angle Calculation for the **ADUX1020**

EVENT DETECTION AND GESTURE WINDOW ALGORITHM

In this section, the pseudocode for the following algorithm descriptions are based on MATLAB syntax. The loops, case statements, and functions follow standard MATLAB format and are derived from actual MATLAB code.

The [ADUX1020](#) evaluation software supports five unit gestures: swipe up, swipe down, swipe left, swipe right, and click. The algorithm waits for an event to occur before classifying data into one of these gestures.

Prior to event detection, the offsets of each of the four channels must be removed, either by subtracting them digitally or by modifying the offsets of the device. Digital removal of the offsets is the simplest way to manage them; however, modifying the offsets of the device maximizes use of the dynamic range of the analog-to-digital converter (ADC) and is recommended in cases where high dynamic range is required. Register 0x3A, Register 0x3B, Register 0x3C, and Register 0x3D contain the ADC offsets, which are nominally set to 0x2000. To modify these offsets, measure the 16-bit output of each channel, in ADC codes, and subtract it from the 16-bit number in the ADC offset register, CHx_OFFSET (nominally 0x2000). Then, write the ADC offset register with the result of this subtraction.

The start of an event can be defined as occurring when either the intensity data or its time derivative crosses a preset threshold. In this application note, a method using the derivative of intensity is utilized, but both methods can be useful.

The intensity data is low-pass filtered to remove high frequency noise. Typically, removal of high frequency noise is completed with a second-order, Butterworth low-pass filter with a 10 Hz corner frequency.

$$I_F = LPF(I)$$

Calculate the derivative of filtered intensity by taking the difference of the successive filtered intensity samples.

$$\frac{dI}{dt}(n) = I_F(n+1) - I_F(n)$$

The start of an event occurs when the derivative of intensity, dI/dt , rises above a preset threshold, I_{TH} , which is in units of ADC codes. Nominally, this threshold is set to 200 ADC codes but can be adjusted up or down to eliminate smaller magnitude to prevent undesired events from being classified as gestures, or to include smaller events as gestures. The gesture start can be found by sweeping through each successive value of dI/dt until the threshold is reached.

For $n = 1:\text{numel}(dI/dt)$

if $dI/dt(n) > I_{TH}$

gestureStart = n

break

Find the end of the gesture by sweeping backward through the dI/dt data, starting from the gesture start time plus the maximum gesture duration, until the derivative of the filtered intensity falls below $-I_{TH}$. Examples of the determination of the start and stop times for a gesture are shown in Figure 3 and Figure 4.

For $n = (\text{gestureStart} + \text{gestureDuration}):-1:\text{gestureStart}$

if $dI/dt(n) < -I_{TH}$

gestureStop = n

break

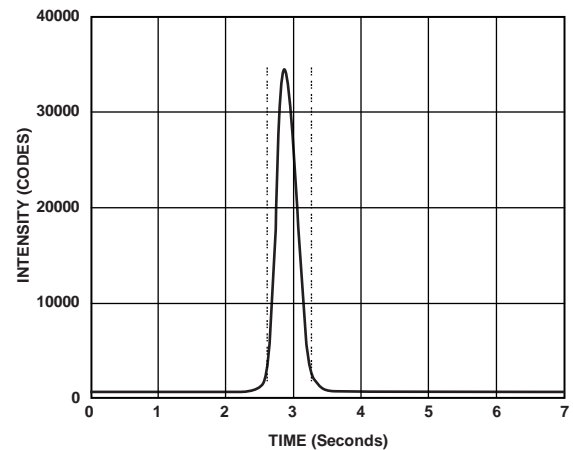


Figure 3. Intensity vs. Time with Gesture Start and Stop Times

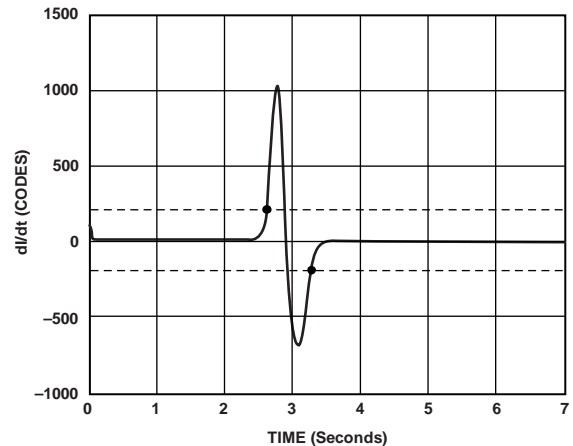


Figure 4. dI/dt vs. Time with Gesture Start and Stop Times

When the gesture is delineated with a start and stop time, the maximum and minimum θ_x and θ_y within the gesture can be calculated. If both quantities, $\max(\theta_x) - \min(\theta_x)$ and $\max(\theta_y) - \min(\theta_y)$, are less than the threshold of θ_{TH} , the event is classified as a click. If the event is not a click, further processing is required to determine the direction of the gesture. A linear least squares fit line between θ_x and θ_y determines whether the gesture is in the category of up and down or left and right, using the following equation:

$$\theta_y = m\theta_x + b$$

where m is the slope.

If the absolute value of the slope of this line, m , is less than 1, the gesture is left and right. If the absolute value of the slope is greater than 1, the gesture is up and down. Note that a basic intensity threshold can also delineate gesture start and stop times for computational simplicity.

If $\max(\theta_x) - \min(\theta_x) < \theta_{TH}$ and $\max(\theta_y) - \min(\theta_y) < \theta_{TH}$

$gesture = click$

else

$(m, b) = \text{leastSquaresFit}(\theta_x, \theta_y)$

if $m \leq 1$

if $\theta_x(gestureStart) > \theta_x(gestureStop)$

$gesture = left$

else

$gesture = right$

else if $m > 1$

if $\theta_y(gestureStart) > \theta_y(gestureStop)$

$gesture = up$

else

$gesture = down$

An example of the least squares fit line is shown in Figure 5. This gesture has a significant spread of θ_x values and is not a click. The slope of the least squares fit line is < 1 , and $\theta_x(gestureStart) > \theta_x(gestureStop)$; therefore, it is a left swipe.

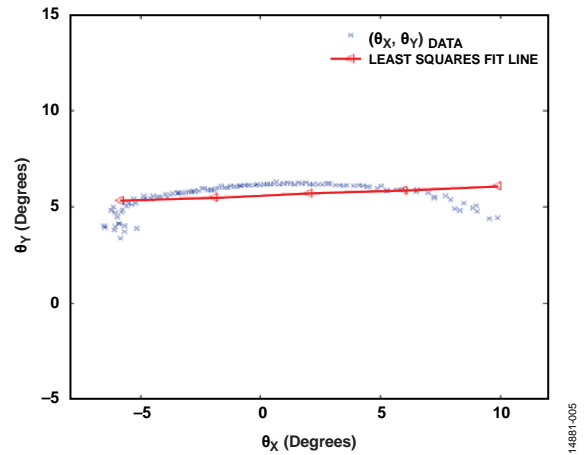


Figure 5. θ_y vs. θ_x with Least Squares Fit Line

Adjusting the parameters of this method ($-I_{TH}$, θ_{TH} , and $gestureDuration$) to refine the algorithm for any use case results in only one possible method of gesture recognition. There are other methods to achieve gesture recognition with the [ADUX1020](#) as well. If the computational load is a concern, only consider the gesture start and stop points can be considered, and ignore those in between. This simple method is very robust and effective, but it requires that the intensity or intensity derivative threshold, I_{TH} , be set high enough to ensure that the start and stop points have adequate signal noise ratio.

Figure 6 shows the intensity dI/dt , and gesture plots for a swipe gesture as seen in the optical gesture evaluation tool. Figure 7 shows the unit gesture recognized as a right swipe.

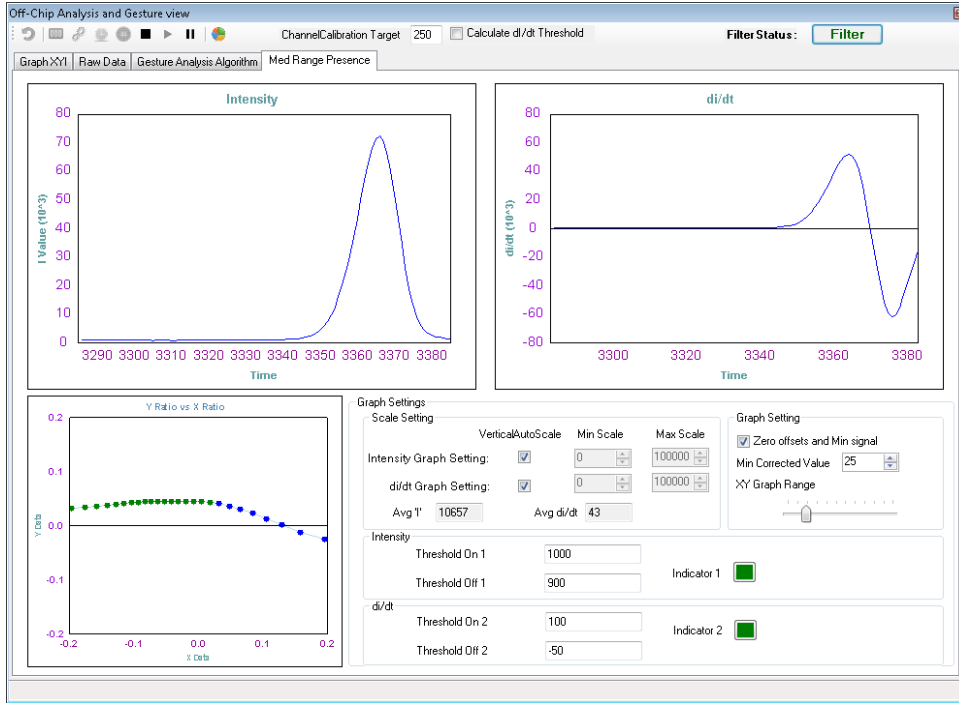


Figure 6. Gesture Recognition Data (Intensity di/dt, and Gesture Plots for a Swipe Gesture) with the Optical Gesture Evaluation Tool

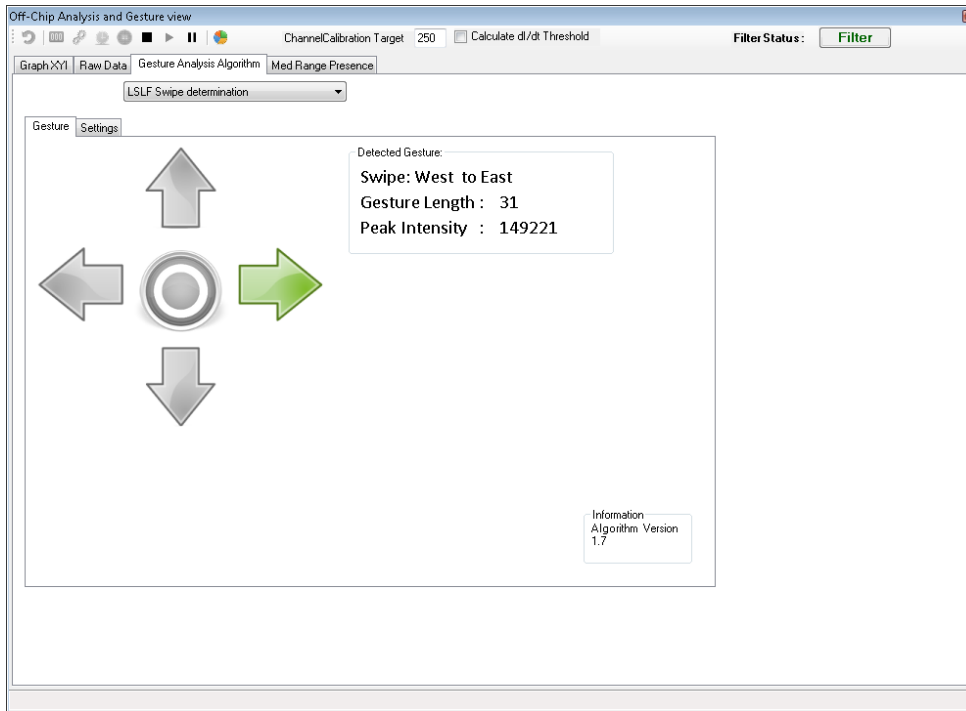


Figure 7. Gesture Swipe Result (Recognized as a Right Swipe) with the Optical Gesture Evaluation Tool