Infrared Gesture Sensing

1. Introduction

Touchless user interfaces are an emerging trend in embedded electronics as product designers seek out innovative control methods and more intuitive ways for users to interact with electronics. Active infrared proximity motion sensing can solve this challenge. Silicon Labs Si114x proximity and ambient light sensor products are ideally suited to touchless gesturing applications such as page turning on an e-reader, scrolling on a tablet PC, or GUI navigation. The Si114x features up to three LED drivers and has the ability to sense gestures within a 7 to 15 cm product interaction region, assuming a hand as the detectable object. This document will discuss in detail how Silicon Labs implements motion sensing using infrared technology. There are two primary methods used for gesture sensing – position-based and phase-based. Position-based gesture sensing involves finding gestures based on the calculated location of an object while phase-based gesture sensing is based on the timing of the changes in signal to determine the direction of an object's motion.

2. Hardware Considerations

This application note focuses on detecting gestures made by a user's hand. It is important to recognize that the concepts introduced in this application note can be applied to targets other than the hand, as long as the hardware is designed appropriately. The end application and individual system constraints will each dictate the range requirements for IR gesture sensing. Since object reflectance is the main measurable component for touchless gesturing, a hand is presumed to be the detectable object for the examples in this document. Whereas a hand can achieve gesture sensing up to 15 cm away from the Si114x sensor, a finger, with dramatically lower reflectance, can achieve gesture sensing at a range of < 1 cm for thumb-scroll type applications.

The general guideline for designing a gesture sensing system with multiple LEDs is to make sure that there is no "dead spot" in the middle of the detectable area. When a target is placed above the system and is not detected, the target is in a reflectivity dead spot. To avoid dead spots in the system, the LEDs must be placed such that the emitted infrared light can reflect off the target and onto the sensor from the desired detection range. Figure 1 shows systems designed to detect a hand or a finger. The most susceptible area for a dead spot is directly above the sensor in between the two LEDs. The two LEDs are placed as close to the edge of the target as possible to provide feedback in the middle while also keeping enough distance between the LEDs so that it can be detected when the finger or hand moves to the left or right.

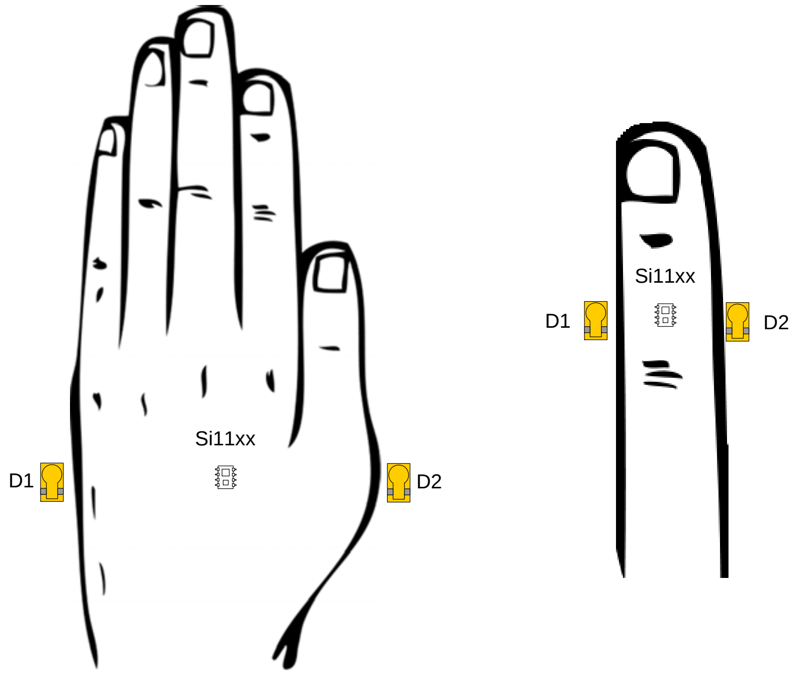


Figure 1. Ideal Hardware Formations—Hand versus Finger Detection

Other than the placement of the LEDs in relation to the size of the hand, the location and reflectance of the target in relation to the system is also very important. Notice that the Si11xx device is placed under the palm of the hand and middle of the finger. For the hand detecting system, the fingers are not a good focal point because light can slip between the fingers, and the shape/curve of the outline of the fingers also makes for unpredictable measurements. For the finger detecting system, the tip of the finger is curved and reflects less light than the middle of the finger.

For more information about other hardware considerations for infrared proximity sensing in general, refer to "AN498: Si114x Designer's Guide". For more information about LEDs, refer to "AN521: irLED Selection Guide for Si114x Proximity Applications".

2.1. Method 1—Position-based Gesture Sensing

The position-based motion sensing algorithm involves three primary steps. The first step is the conversion of the raw data inputs to usable distance data. The second step uses the distance data to estimate the position of the target object, and the third step checks the timing of the movement of the position data to see if any gestures occurred.

- **Step 1**—**Converting Counts to Distance**: The infrared (IR) proximity sensor outputs a value for the amount of infrared light fed back to the device from the IR LEDs. These values increase as more light is reflected by an object or hand moving closer to the system. This example presumes that a hand is the defined target for detection. The system can estimate how far one's hand is based on characterization of the proximity sensing (PS) feedback for a hand at certain distances. For example, if a hand is placed in front of the system about 10 cm away and the PS measurement yielded 5000 counts, then subsequent PS measurements around 5000 counts also mean that there is a similarly reflective hand approximately 10 cm away from the system.

  Taking multiple data points at varying distances from the system helps to interpolate between these points and create a piece-wise equation for the conversion from raw PS counts to distance estimation. Note that for systems with multiple LEDs, each LED will have a different PS feedback for each hand distance so each LED will need its own counts to distance equation.

In Figure 2, the characterization for a two-LED system is displayed. Each LED must be characterized with a target suspended over the midpoint between the LED and the sensing device. When Target 1 is suspended over the sensing device and IR LED 1, the measured feedback will correlate to a distance D1 above the system. The same is true for Target 2, IR LED 2, and D2.

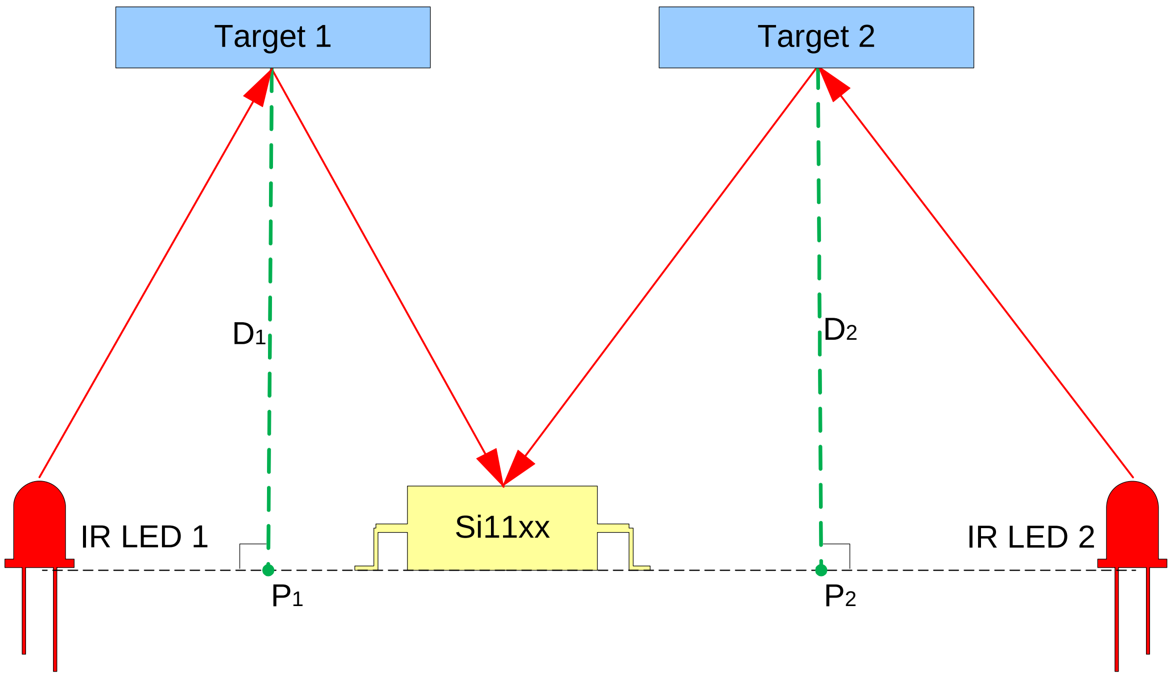


Figure 2. Distance Approximation

- **Step 2—Estimating Position**: The next step in the process of detecting gestures is to take the distance data from Step 1 and estimate the target's position. The position estimate comes from the formula for the intersection of two circles. An LED's light output is conical in nature, but for this approximation, the LED's output is considered to be spherical. With the LED's on the same axis, the intersection of these two spheres can be considered using equations for the intersection of two circles.

  Consider the characterization performed in Figure 2 in the previous section. Figure 3 shows what the estimations D1 and D2 mean when a single target is suspended over the middle area of the system. D1 and D2 are estimations of the distance from the points P1 and P2 to the target above the system. If D1 and D2 are thought of as radii of two circles, the intersection of these two circles is the point where the target is located.

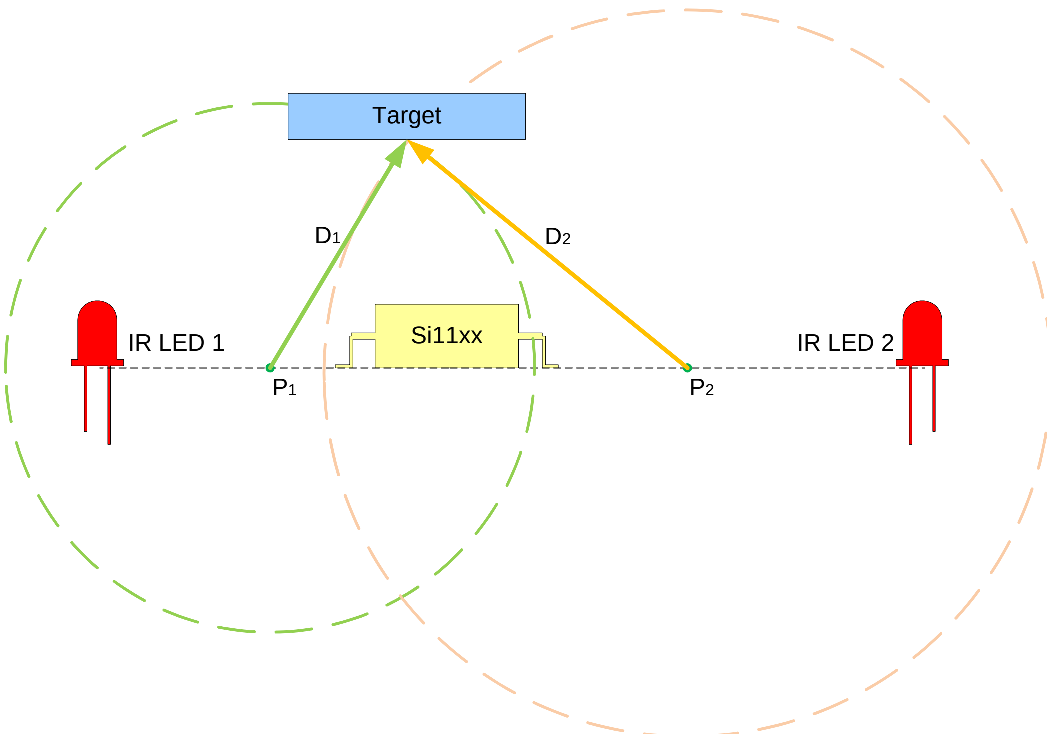


Figure 3. Single Target Distance Approximation

Figure 4 is an expanded version of Figure 3. The measurements "a" and "b" have been added to label the location of the target along the axis between the two points P1 and P2. Also note that the distance measurements D1 and D2 have been renamed to R1 and R2 to represent that they are being considered as radii now.

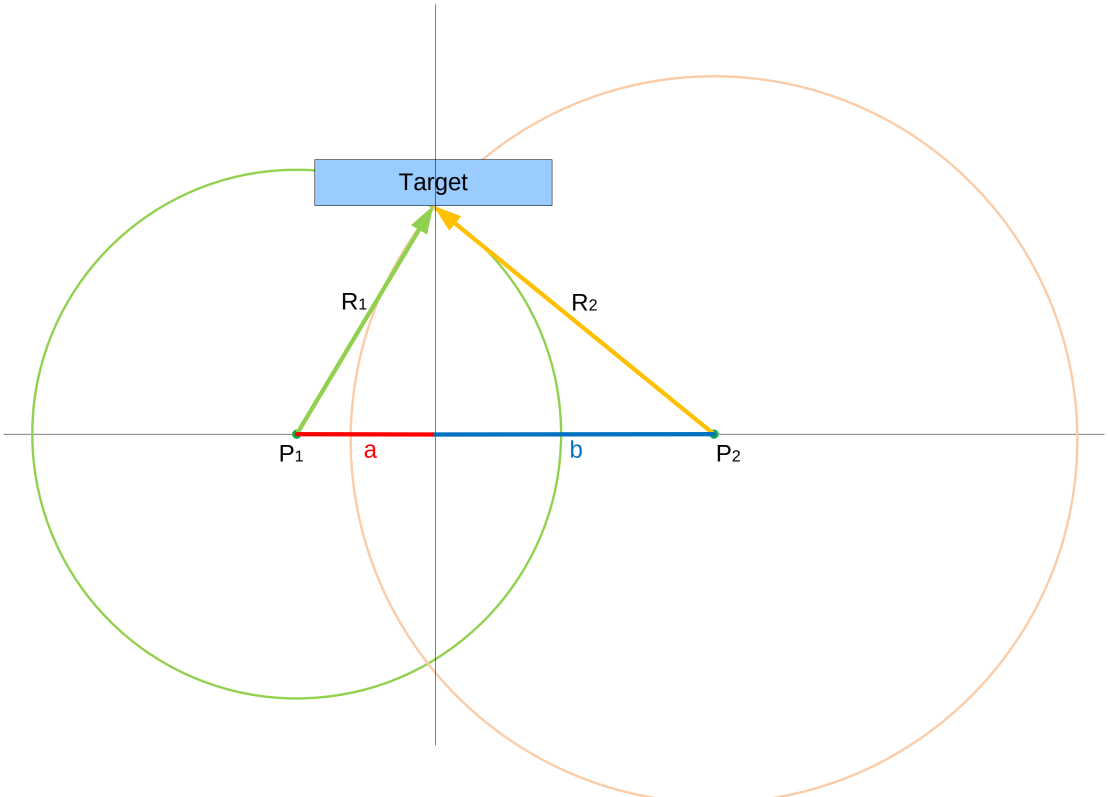


Figure 4. Intersection of Two Circles Implementation

$$\text{Using } d = a + b,\ a = (R1^2 - R2^2 + d^2)/(2 \times d)$$

The value of "a" will be the location of the object along the axis between P1 and P2. A negative value is possible and will mean that the target is on the left side of P1.

- **Step 3—Timing Gestures**: With a positioning algorithm in place, keeping track of timing allows the system to search for and acknowledge gestures. For hand-swiping gestures, the entry and exit positions are the most important to consider. If the hand enters the right side with a very high "a" value and exits the left side with a very low "a" value, then a left swipe occurred, as long as the timing between the entry and exit happened within a chosen time window. If the position stays steadily in the middle area for a certain period of time, then this gesture can be considered a pause gesture.

  The system must keep track of the timestamps for the entry, exit, and current positions of the target in the detectable area. With timing information and position information, most gestures can be recognized. Timing will need to be custom tuned to each application and each system designer's preference. Silicon Labs Reference Designs use a quarter of a second for the gesture time requirement.

  Silicon Labs' IrSlider2EK is a great example and starting point for comprehending the fundamentals of position-based gesturing.

2.2. Method 2—Phase-based Gesture Sensing

For the second method of gesture sensing, the location of the target is never calculated. This means that Steps 1 and 2 from Method 1 above are not used in this process. This method involves looking solely at the raw data from the proximity measurements and looking for the timing of the changes in feedback for each LED. The maximum feedback point for each LED will occur when a hand is right above that LED. If a hand is swiped across two LEDs, the direction of the swipe can be determined by looking at which LED's feedback raised first. Figure 5 shows a three-LED system with a hand about to swipe over the system.

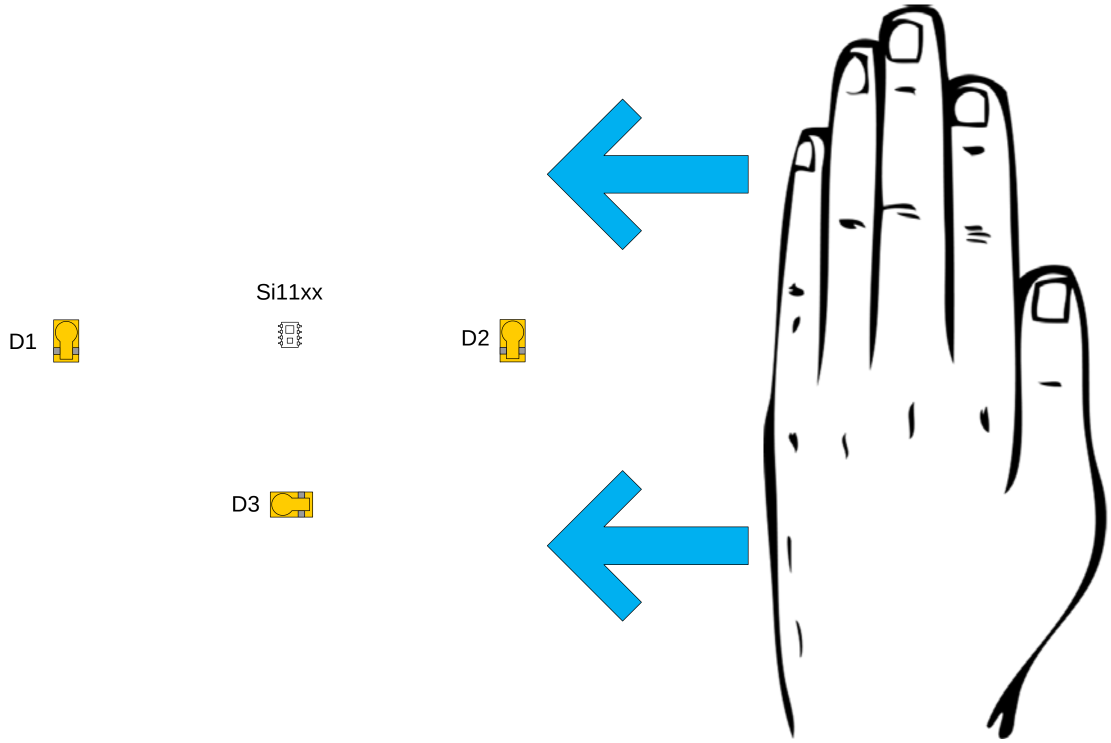


Figure 5. Swiping Left over a 3-LED System

When the hand in Figure 5 is swiped left across the three LEDs and proximity sensing device, the hand will first cross over D2, then D3, then D1. The sensing algorithm will recognize the rise in feedback for D2 and record the timestamp for this raise. Then the sensing algorithm will detect the same raises for D3 and D1 with a higher timestamp than the one before it. The sensing algorithm can also detect the return back to no-detect state of each LED's measurement and record a timestamp for this as well. First D2 will return back to normal, then D3, then D1. For up and down gestures in the system in Figure 5, D1 and D2 will rise and fall simultaneously with D3 coming either before or after D1/D2 for the up or down gesture. If all three channels rise and fall at the same time, then a hand approached from directly above (Z-axis) and was retracted to indicate a "select" gesture.

In Figure 6, the four gestures right, left, down, and up are shown on graphs of ADC Counts versus time. The green line represents PS measurements using D1, the pink line represents PS data from D2, and the yellow line shows data from D3. Observe for the Right Swipe that D1 spikes first, then D3, and finally D2. This is expected for a swipe from left to right. For the up and down swipes, D1 and D2 spike at the same time since your hand will cross these LEDs at the same time when swiping up or down.

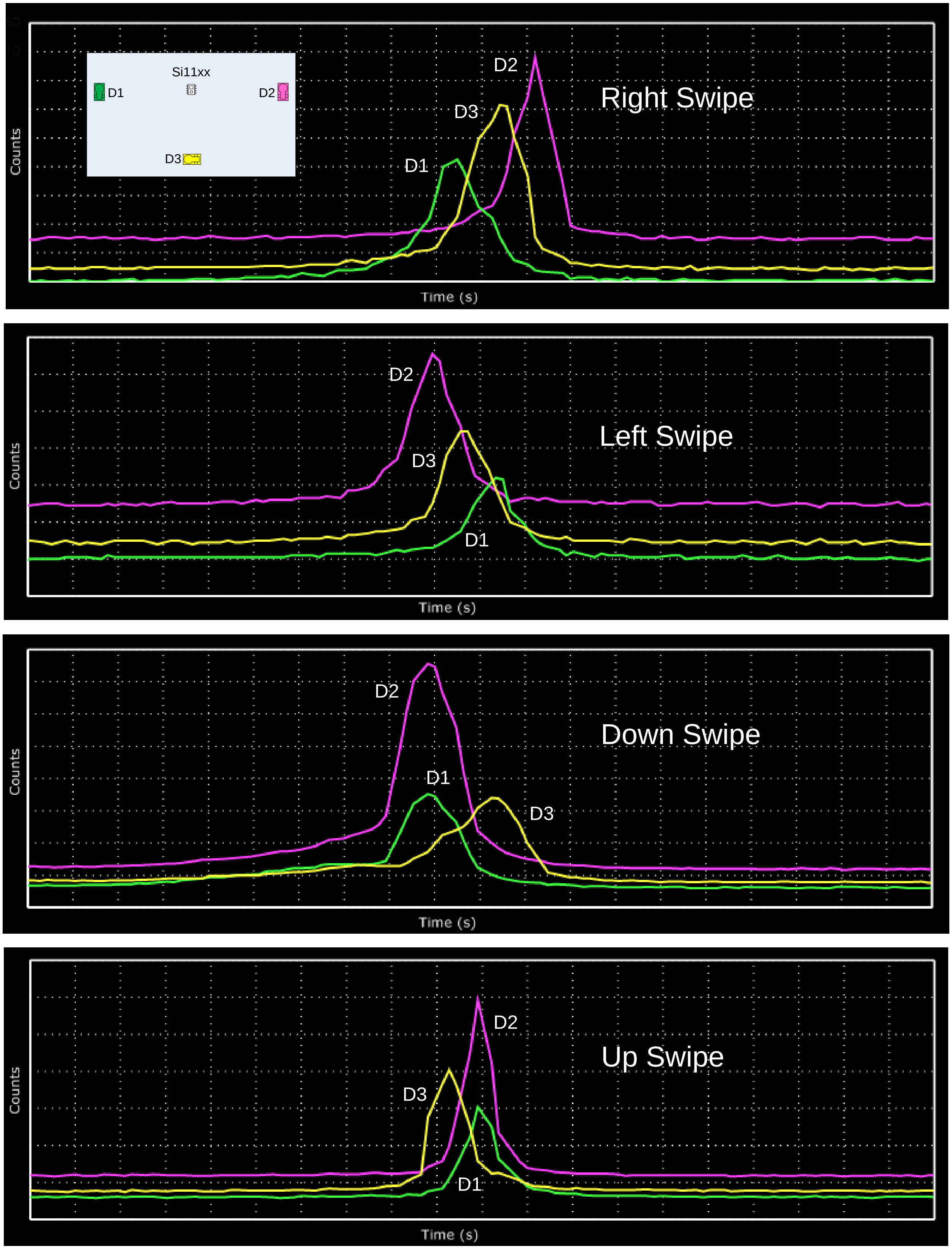


Figure 6. Gesture Data

3. Position-based Method Advantages and Drawbacks

The advantage of the position-based method over phase-based is that the position-based method can offer information on the location of the target. This will allow for ratiometric control of systems. For example, to scroll through a book several pages you could suspend your hand over the right side of the detectable area instead of having to make several right swipe gestures.

The main drawback of the position-based algorithm is the accuracy of the position calculations. The positioning algorithm assumes a spherical output from the LEDs, but in practice LED output is more conical than spherical. The algorithm also assumes uniform light intensity across the entire output of the LED, but the light intensity decays away from the normal. Another issue is that this algorithm does not account for the shape of the target. A target that is uniquely shaped will cause inconsistencies with the positioning output. For example, the system cannot tell the difference between the hand and the wrist, so any gestures involving movement that puts the wrist in the area of detection will be less accurately located.

The result is that the positioning information provided in this algorithm is good enough for low resolution systems that only need a 3x3 grid of detection, but the current positioning algorithm is not well-suited for a pointing application. This algorithm's output is not an ideal touchscreen replacement.

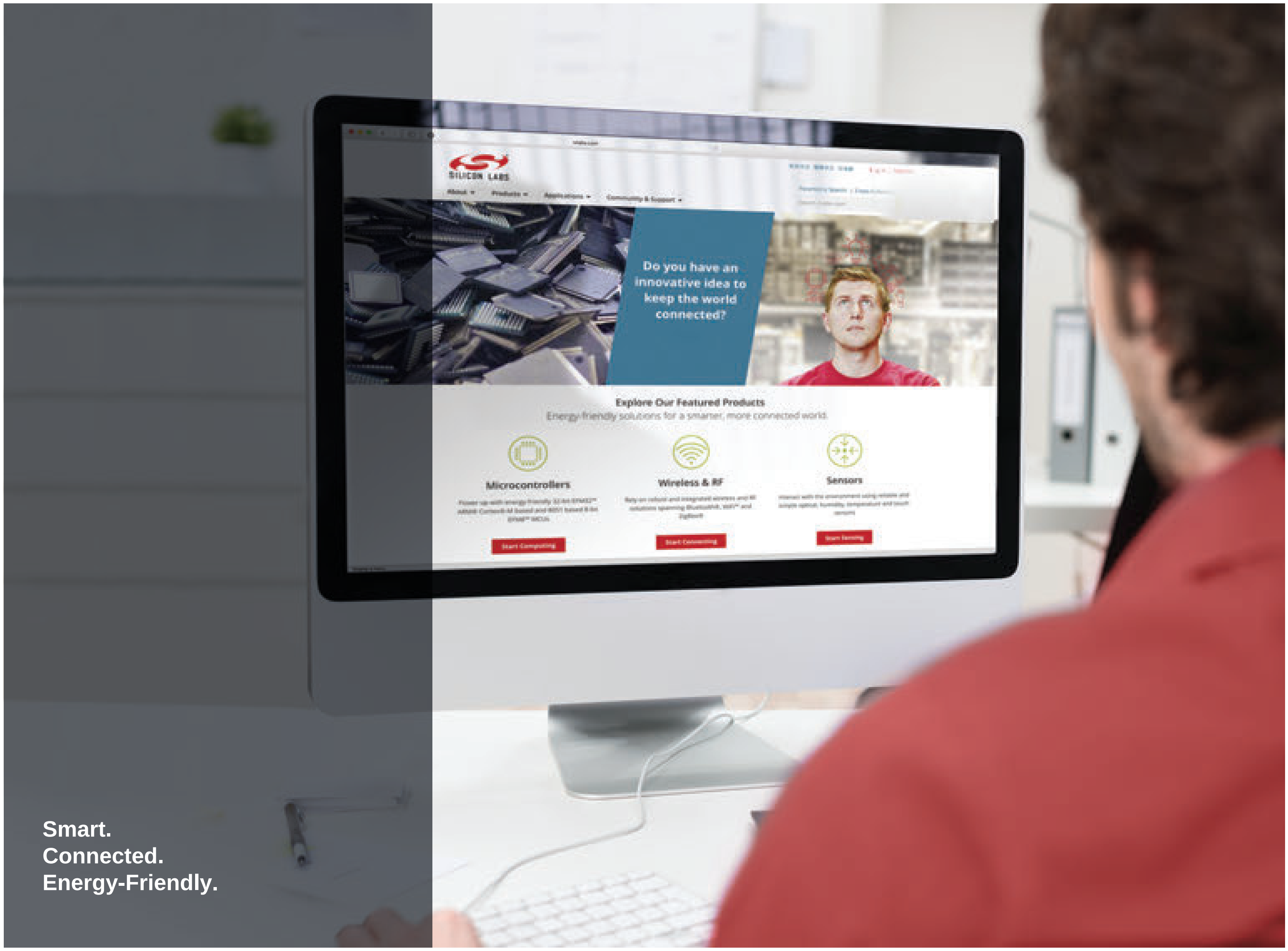
4. Phase-based Method Advantages and Drawbacks

For applications not requiring position information, the phase-based method provides a very robust way of detecting gestures. Each gesture can be detected on either the entry or exit from the detectable area, and the entry and exit can be double-checked with each other to provide much higher certainty for each gesture observed.

The drawback of this method when compared with the position-based method is that no positioning information is provided. This means that the number of gestures that can be implemented are more limited than the position-based method. The phase-based method can only tell the direction of entry and exit from the detectable area so any movement in the middle of the detectable area is not detected.

5. Improved Gestures through Combination of the Two Methods

These two methods can be implemented alongside one another to help mask the other's deficiencies. The position-based algorithm can provide some positional information for ratiometric control, and the phase-based algorithm can be used for detection of most gestures. These two algorithms working together can provide a strong solution for gesture sensing, but the drawback here is the code space requirements of implementing two separate algorithms as well as the CPU cycles to process both algorithms at all times.



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com