Gesture control allows users to interact with electronics in intuitive ways without having to touch the device they are controlling. This creates a new user experience in which commands can be more conveniently and quickly executed.

**ADVANTAGES**

There are many advantages of using gesture control over capacitive touch control or more traditional control devices like buttons or knobs. When wearing gloves, capacitive touch panels will not work, but infrared gesture control will. While both sensing technologies work in two dimensions, x and y, gesture control adds a third dimension to the control mix. Users can move their hand towards and away from the sensor to execute control. In applications where safety might be compromised by requiring a user to find a knob or push a specific button, gesture control allows a user to make these changes in a wide active field where gross movements can replace finer movements. Simple, more natural hand movements can replace less intuitive control technology. For example, a radio channel can be changed with the simple left or right swipe of a hand. Volume can be controlled by an up or down swipe or a push in or out motion. Finally, no stylus or input device is required other than your hand. Gesture control is not a panacea for all user controlled devices but it can augment existing technology and thus bring added value to a consumer.

**APPLICATIONS**

The use of gesture control in its crudest form is already a part of our everyday lives. Proximity sensors used for soap dispensing, hand towel dispensers, and automatic flush and faucets may not require a specific gesture but they are an embryonic form of gesture control. More sophisticated gesture control has been integrated into tablet PCs and smartphones where left and right swiping is commonly featured. While not widely available yet, light switches and dimmers are being designed with infrared gesture control, replacing mechanical switches. Automotive manufacturers are looking for ways to add convenience and increase safety for drivers by integrating gesture control into radio and climate control and other body electronics. Microsoft’s Kinect, the Wii, and the LeapMotions gesture device are the most sophisticated gesture control devices commercially available. But because they use lasers, CCD cameras, and complex photo detectors, these devices tend to be very expensive and software intense compared to VCNL4020-based gesture control.

If you pay attention throughout the day to the number of times you have to touch something to cause an effect, you will find many examples where infrared gesture control could be used to replace touch. The VCNL4020 Gesture Control Sensor Board is the ideal tool to quickly prove a design concept and to start initial hardware and software gesture control development.
VCNL4020 Proximity Sensor - Gesture Control Sensor Board

SENSOR BOARD
The foundation of Vishay’s Gesture Control Sensor Board is the VCNL4020 proximity and ambient light sensor. Two of Vishay’s VSMF2890RGX01 infrared diodes are mounted on either side of the sensor. They are mounted on the opposite side of the PCB from the VCNL4020 sensor, minimizing cross talk between the diodes and the sensor. A high radiant intensity of typically 80 mW/sr at 200 mA allows for the detection of hand gestures up to 25 cm above the sensor board. An image of the board is shown below.

GESTURE RECOGNITION
The detection of a gesture is composed of two main parts: The acquisition and preparation of the raw data to be analyzed and the interpretation of this raw data by the detection algorithm. If a gesture is recognized, a corresponding event is triggered.

Data Acquisition
The data used for gesture recognition is acquired in separate data streams, one per emitter. As the emitters are multiplexed (and only one emitter should be on at a time), the measurements occur slightly one after the other. One measurement cycle consists of a set of data from each channel. The measurements are done in quick succession so that the time required to complete a full measurement cycle is much greater than the time between the individual proximity measurements of each IRED. In this way the signals can be compared with one another in close to real time.
If the jumps between measurements are seen to be too erratic, for example if the sensor is in a high noise environment, several measurements per emitter and per measurement cycle can be made and averaged. This will slow down the measurement speed, but it will result in a cleaner signal. This variable can be adjusted in the provided gesture demo software, as will be discussed later.

To be able to analyze and compare the data streams of each emitter channel with one another, the data of each stream is split into frames of “n” measurements. Here, each frame contains the latest “n” measurements, with each iteration of the measurement cycle, moving the frame over by one. The recognition algorithm analyzes each frame and once a frame is seen to contain a gesture, the next few frames are ignored, in order to avoid detecting the same gesture twice. Furthermore, by ignoring several frames after a valid gesture has been detected, it is possible to reduce false positives from hand movements in the sensor’s field of view that were not intended as a gesture. This variable can be adjusted in the provided gesture demo software.
Detection Algorithm

Each frame of the acquired signal is analyzed for two parameters: the standard deviation of each signal and the time delay between signals. By comparing the results of this analysis to user-defined thresholds, the algorithm can tell whether and what kind of a gesture was made.

The standard deviation is a measure of the spread of the data within the frame being analyzed. It is calculated using the following formula:

$$s = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

where $\bar{x}$ is the mean of the current frame and $n$ is the amount of samples being analyzed, i.e. frame length.

A low standard deviation implies there is no change in the signal and there is either no hand in the sensor’s detection area or the hand is held steady over the sensor and no swipe or push gesture is being made. A high standard deviation implies a large change in the signal, suggesting the movement of the hand across or towards the sensor. The detection algorithm only analyzes the frame for further parameters, if the signal is above a set standard deviation threshold.

The presence of sufficient time delay between the signals signifies that a swipe gesture has been made. In the detection algorithm implemented in the demo kit, the time delay is found by measuring the cross correlation between the two signals. The cross correlation can be calculated by the following formula:

$$\text{corr}(f[x], g[x]) = \sum_{k=\infty}^{\infty} f[k]g(x + k)$$

where $f[x]$ and $g[x]$ are individual data streams from either IRED and $f^*$ denotes the conjugate of $f$.

The cross correlation effectively shifts one of the signals over the other and computes the percent overlay of their integrals. By taking note of the point at which there is a maximum cross correlation between the two signals, an estimation of the time delay between the two signals can be found. As the cross correlation of the signals is always computed in the same order, regardless of which of the signals comes first, the time delay will either be positive or negative. The sign of the time delay denotes the direction of the swipe gesture. The decision on whether a right or left swipe was detected is only made when this time delay threshold is exceeded.

Both the time delay threshold and the standard deviation threshold can be adjusted in the demo software, allowing the designer to test and fine tune for their given application.

The method described above is very sensitive and robust, but there are other, less computationally intensive ways to detect gestures. For example, one can monitor a threshold over which hand entry and hand exit events are detected. An initial rising edge of the threshold of a channel denotes the presence of a hand over that emitter, and a falling edge means the hand has left the detection area. Depending on the event order, a differentiation can be made between a left and a right swipe. This method has no need for framing, and instead analyzes the result of each measurement cycle one by one. This approach may be more suitable for some applications. Such an algorithm was also tested with the Gesture Sensor Board, and the VCNL4020 proved to be just as effective.
GESTURE CONTROL DEMO SOFTWARE

Gesture Function Tab
The Gesture Sensor Board comes with demo software that implements the detection algorithm and visualizes the detection of gestures. The software allows the key parameters of both the data acquisition and the detection algorithm to be changed, in order to see their effect on the system.

The software starts in “Gesture Mode” and the board should be orientated as depicted. The legend allows the sensor board user to differentiate between the two signals, with the color codes matching the color coding of the IREDs in the image. When the measure button is pressed the software starts to make continuous proximity measurements and a movement of the hand above the sensor will cause an increase in counts of the reflected signal, as displayed in the measurement window. If a performed gesture is detected, it is displayed in the gesture window. In order to gain a better understanding of the signals that led the algorithm to recognize the gesture, they are displayed in the gesture signals window.
The right hand side of the Gesture Function tab shows current measurement statistics. Depending on which mode has been selected, the non relevant statistics and options will be grayed out. The “Measure” button allows the data acquisition to be started and stopped, and the “Clear Display” button clears the graphs and resets all displayed valued to their initial state.

There is an option to regulate the forward current with which each of the IREDs is pulsed. This allows the signals to be better matched in case of intensity deviations. The figure below shows the current of IRED2 being decreased to match the signal levels.
Automatic Offset
The measurement window signal shows the actual counts being read. In order for the gesture algorithm to function correctly, an automatic offset calibration is built in. Here the offset is compensated in relation to the average of the first 10 measurements after the Measure button has been pressed. The user should wait until the first 10 measurements have been made before executing a gesture, as the detection algorithm is paused during this time. This allows a cover glass to be placed over the board before the measurement button is pressed and gestures will still be detected.

Setup Tab

The main parameters that impact both the acquisition of the data and the thresholds that the gesture algorithm uses to determine the occurrence of a gesture are presented in the Setup tab.

**Modes of Operation:**
- **Both:** Both IREDs pulsed at the same time for an increased detection range. This can allow for a type of “Wake up” function
- **IRED1/IRED2:** Pulses one of the two IREDs, allowing the board to function as a proximity sensor with one external IRED
- **Gesture:** Pulses the IREDs in turn, generating a data stream per IRED. This data is cut into frames and then analyzed by the gesture recognition algorithm, as discussed above.

**Parameters for Data Acquisition:**
- **Delay Time:** The time that the software pauses between measurement cycles, determining the sampling rate. The smaller the delay time the more accurately the time delay can be measured. This number should be kept low enough to allow for faster gestures to be detected.
- **Number of Frames to Skip:** When a gesture is detected, the gesture detection algorithm is deactivated for “n” frames. This number should be at least “1” so that the same signal cannot lead to more than one detected gesture. The more frames that are skipped, the longer the wait time between gestures. If multiple gestures are to be detected in quick succession, this number should be set low. If there are false positives after a gesture, this number should be increased.
VCNL4020 Proximity Sensor - Gesture Control Sensor Board

- **Frame Length**: Determines the amount of measurements per frame. The frame should contain enough measurements to capture the full signal of one gesture, which also depends on the speed at which the gesture is executed. A longer frame length will allow very slow gestures to be detected, as more data points are analyzed at one time. The longer the frame length the more measurements are ignored after a gesture has been detected.

- **Reading per IRED**: Determines the amount of measurements that are made and averaged for each data point, per measurement cycle. Increasing this number will lead to a more jagged signal. If the signal is seen to have a high peak to peak noise, increasing this number can decrease the amount of false positives interpreted by the gesture algorithm. However this “per measurement” averaging takes time, decreasing the measurement speed, making the algorithm less responsive.

**Parameters for Gesture Detection:**
- **Standard Deviation Threshold**: The standard deviation is a measure of the spread of the data within each frame. The movement of the hand over the gesture board will increase the standard deviation of both signals. The detection algorithm will only scan the frames for a time delay between the signals if the standard deviation surpasses this set threshold. This threshold should be set higher than any seen signal noise. The further away from the sensor a gesture is performed, the smaller the change in standard deviation, as fewer counts are detected. The gesture detection range increases with a lower standard deviation threshold, as less change in the signal is required to exceed the threshold.

- **Time Delay Threshold**: The threshold determines the minimum required time delay that needs to be detected by the gesture algorithm, for the algorithm to execute a gesture event and analyze if it was a negative (left swipe) or positive (right swipe) delay. The detectable time difference will depend on the measurement rate. With a lower measurement rate (higher delay time) the measurement time resolution decreases and so the time delay threshold must be decreased. The width of the object used to perform the gesture should also be taken into consideration. A thinner object will lead to a higher detected time delay between signals, whereas a wider object will lead to a lower detected time delay.

An example of a successfully detected left and right swipe can be seen below:

The Gesture Sensor Board is an add-on board to the VCNL4000 Demo kit. More information about this demo kit can be found at www.vishay.com/docs/83395/vcnl4000_demo_kit.pdf.

Once you have purchased a VCNL4000 demo kit please contact sensorstechsupport@vishay.com if you wish to receive a Gesture Sensor Board.
The switching information (IRI and IRE) is delivered from the USB controller and follows the specification shown below:

<table>
<thead>
<tr>
<th>IRI</th>
<th>IRE</th>
<th>IRED OPERATING</th>
</tr>
</thead>
<tbody>
<tr>
<td>L or open</td>
<td>L or open</td>
<td>IRED_D2</td>
</tr>
<tr>
<td>L or open</td>
<td>H</td>
<td>forbidden</td>
</tr>
<tr>
<td>H</td>
<td>L or open</td>
<td>both IREDs</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>IRED_D3</td>
</tr>
</tbody>
</table>