Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

By John Fisher and Anika Kühnle

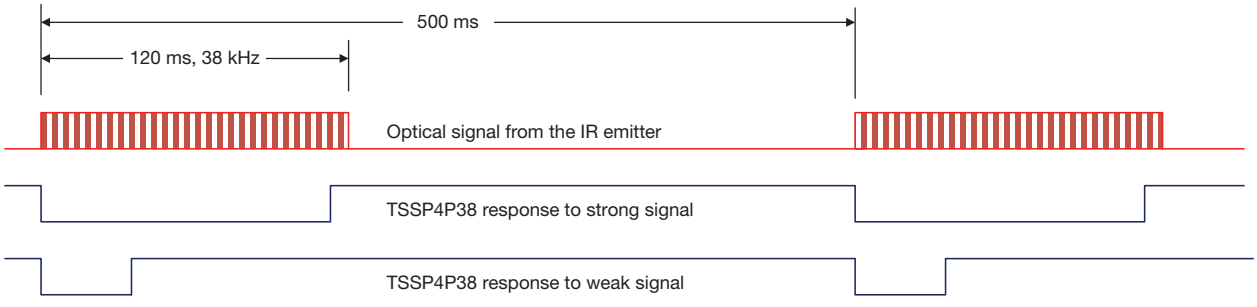
Vishay's TSSP sensor series was designed to fulfill the need for low-cost, simple-to-use reflective and beam break sensors. These sensors use modulated infrared (IR) light to achieve a high range, up to 2.5 m for a reflective sensor and up to 30 m for a beam break sensor. The TSSP devices function only as the receiver in these sensor solutions; a separate IR emitter must be used to generate the signal. For a selection of suitable IR emitters, please consult the product guide here: www.vishay.com/doc?49495.

The TSSP series contains two main types based on their automatic gain control (AGC) classification. The constant gain or AGC "0" sensors (which provide no AGC) are for applications that require a consistent, well-defined response under any level of ambient light. Variable gain AGC "P" sensors are designed to capture relative proximity information from a moving object using reflected IR light. This application note focuses on the AGC "P" sensors; constant gain "0" sensors will be the subject of a future application note.

AGC P sensors use the strength of the reflected signal to measure relative proximity. It is assumed that in a typical application, the size of the object and its reflectivity to IR remain constant, and that only its distance from the sensor varies. It follows that AGC P sensors are only capable of measuring absolute distance when used with a calibrated object, but they are reasonably accurate at determining relative proximity with random objects. They are also inexpensive and require a minimum number of external components to create a complete solution. In this project, a low-cost Arduino Nano board will be used to generate the emitter signal and to convert the sensor output to a human-readable form.

THE PROXIMITY SENSOR

The TSSP AGC "P" sensors capture relative proximity information by emitting a long burst signal (100 ms to 150 ms) via a separate IR LED, and evaluating the time required by the AGC to internally suppress the reflected signal. The AGC dynamics of these devices can be separated into three distinct phases. The first phase is the pre-triggered AGC state. In this phase, no signal has been detected and the sensor gain is at its maximum. At maximum gain, even weak reflected signals at the modulation frequency of 38 kHz will trigger the AGC, driving the sensor's output pin active low. Once the AGC has been triggered, the AGC reduces the gain of the sensor in the second phase by approximately 100 dB per second until the gain reaches its minimal value. At some level of gain, the signal level internal to the sensor will no longer exceed a pre-set threshold, and the sensor output will return high. It is during this second phase that relative proximity information is determined. The time required to suppress a reflected signal is longer when the signal is strong than when the signal is weak. Therefore, the output pulse is also longer for a strong signal, or nearer object, than for a weak, or more distant object. The gain reduction process continues, even after the reflected signal is no longer present, until the AGC has reduced the gain to its minimum value. In the final phase, the AGC returns the gain to its full value at a rate of about 100 dB per second. It is important that the sensor see no signal during the gain recovery phase as that may prevent it from returning to a known state. A suitable emitter signal must therefore provide for a quiet period long enough for the gain to recover to its full value. Good results have been achieved with a quiet period of at least 380 ms as shown below:



APPLICATION NOTE

Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

THE HOUSING

Another essential part of a reflective sensor is a suitable housing in which to mount the optical components. The housing must not only provide mechanical support, it must guarantee complete optical isolation between the emitter and the receiver. No direct or reflected path may exist between the emitter and the sensor. The gain of the sensor is very high and even small amounts of stray light, e.g. reflected off stray wiring, will activate the sensor and render it blind to reflections from the far field. Care must also be taken to avoid using a common transparent window between the emitter and sensor, which can act as a light guide between the two components.

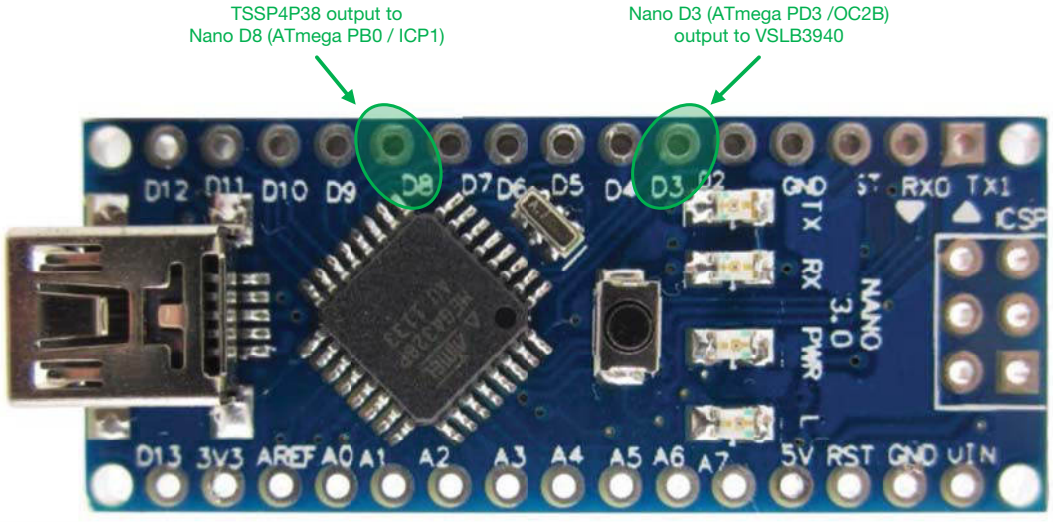
Typically, a housing design is quite specific to the final product and is not available to the designer during the prototyping stage of a project. For this reason, for a limited time Vishay is providing a prototype kit consisting of an injection-mold housing for emitter and sensor, a TSSP4P38 sensor, and a VSLB3940 IR emitter. Please contact our technical support link to request this kit.

Sensor prototyping kit:

TSSP4P38 and VSLB3940 in a plastic housing, providing mechanical support and optical separation.

ARDUINO NANO BOARD

A fast, easy-to-use solution to generate the emitter signal for the VSLB3940 and evaluate the output of the TSSP4P38 is a low-cost development board such as the open-source Arduino Nano depicted below. The heart of this board is an 8-bit ATmega328P microcontroller running at 16 MHz, which is fully sufficient for most IR projects. This board was selected due to its easy insertion in any standard breadboard for prototyping. The Arduino Nano board itself is available from many online sources, and the IDE (Integrated Development Environment) software may be downloaded free of charge from the Arduino website: arduino.cc/en/Main/Software.



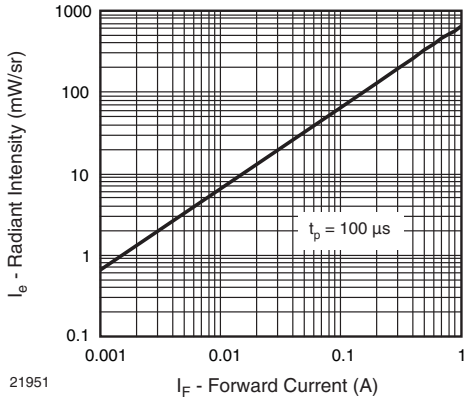
APPLICATION NOTE

Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

A word about pin names: most pins of the ATmega series are multi-purpose and carry different names according to the function that is currently being discussed. The Arduino boards have their own names. In the green text on the picture above, you can see three names for each pin, the Nano's, and in parentheses, the ATmega's name for the common I/O pin, as well as the timer-specific names we are actually using in the software: ICP1 is the Input CaPture pin of Timer1, and OC2B is the Output Compare B pin of Timer2. See the Atmel datasheet for full details.

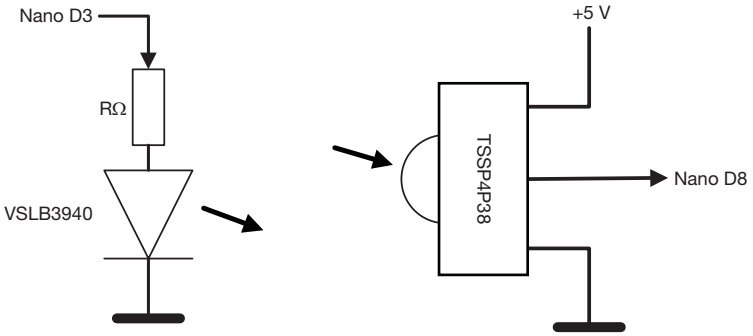
The USB connector allows uploading of the compiled Nano software from the IDE, a means for the Nano to communicate with humans via the serial monitor in the IDE, and also provides the 5 V supply that powers the Nano board as well as both the emitter and the sensor. The TSSP4P38 draws less than 1 mA and may be run from either the 3.3 V regulated voltage on pin 17 or from 5 V on pin 27.

It's far from obvious based on Atmel's datasheet how to calculate the peak I/O pin source currents. The absolute DC rating of each I/O pin is 40 mA, but one must observe the port (a port consists of 7 or 8 I/O pins, depending on the port) maximum currents of 150 mA per port source, and 100 mA per port sink. These are all DC ratings. It is not stated whether pulsed maximum ratings may be scaled higher corresponding to their duty cycle, as long as the average current does not exceed the DC rating. However, 40 mA is enough to drive the VSLB3940 for most sensor applications. The VSLB3940 itself may withstand much higher currents than this, up to 100 mA DC or a whopping 1 A peak. It is important to understand that the transmission distance of modulated IR systems depends on the peak current, not the average. There is a fairly linear relationship between the forward current I_F in an IR emitter and its so-called radiant intensity, I_e. The range is a square root relationship between the radiant intensity of the emitter, and the minimum irradiance parameter of the sensor, $E_{e\ min.}$, which gives the minimum signal power required to get any response out of the sensor. The range for reflective sensors is then $d_{(in\ meters)} = \sqrt{(L_m I_e / E_{e\ min.})}$, where d is the distance from the sensor to the object and back again, and L_m is a reflection loss in the radiant intensity I_e that is material dependent. If very large ranges are needed, the peak forward current to the emitter can be increased beyond the 40 mA limit of the Nano via an external driving transistor.



Relationship between Forward Current and Radiant Intensity in VSLB3940

The component electrical connections are simple:



APPLICATION NOTE

Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

INSTRUCTIONS FOR USING THE IDE

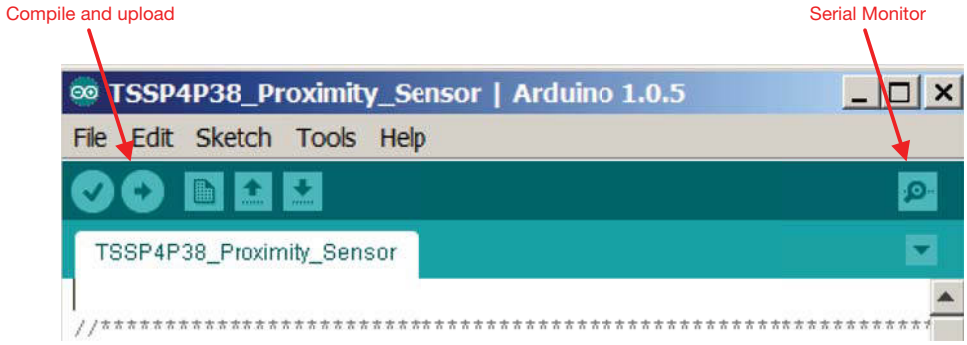
The default installation of the Arduino IDE places the program files in an "Arduino" folder under "My Documents" for Windows or a "Sketchbook" folder under "Home" for Linux. The complete prototype software for this project, written for the Arduino Nano, is available from the TSSP4P38 web page: www.vishay.com/doc?82728 (tssp4p38_proximity_sensor.zip).

Download and unpack the example program to a folder of the name "TSSP4P38_Proximity_Sensor" within the respective program folder on your system. After restarting the IDE, the program may then be opened via the menu File → Sketchbook. The Nano board must be selected under Tools → Board (Arduino Nano w/ATmega328). On Linux, your user name must belong to the "dialout" group in order to use the Serial Monitor feature in the IDE. If you do not see "dialout" when you run *groups* in the command terminal, then running the following command will add you to that group. It's necessary to log off and back on to your system before it will take effect:

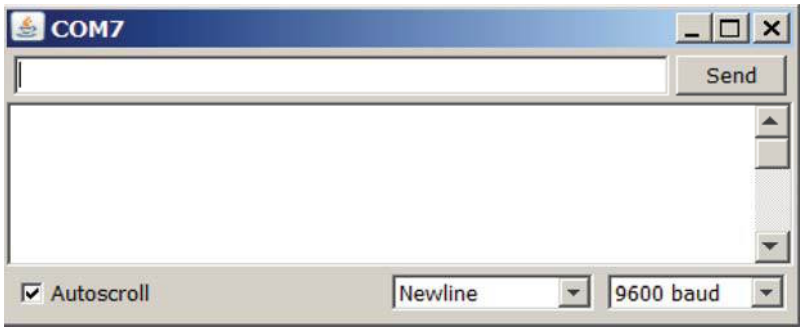
sudo usermod -aG dialout <UserName>

Finally, you must also select the COM port the Nano is using via the menu Tools → Serial Port. Plugging the board in and out with the menu open can help you find the correct one.

Assuming no glitches up to this point, you should be able to open the program TSSP4P38_Proximity_Sensor under menu File → Sketchbook. Compile and upload the program with the icon button showing a right pointing arrow. Next, click on the icon at the right side of the icon bar to open the Serial Monitor.



To be compatible with the sensor program, the Serial Monitor should be set for a baud rate of 9600 and should send Newline characters at the end of each line. See where to set these in the graphic below.



Closing and opening the Serial Monitor resets the Nano and it will prompt you to input the burst length (say 150) and repetition time (500 is good) of the emitter signal, and a "1" to activate the sensor. The monitor should immediately start reporting the sensor's output pulse widths in milliseconds. You may then point the sensor at objects at various distances to check that the pulse widths do indeed change according to their relative proximity.

Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

THE SOFTWARE

The AT Mega's onboard hardware timers make a simple task of creating PWM bursts for the emitter and measuring the resulting sensor output pulse widths. The AT Mega has three hardware timers, counter / timer 0 and 2, both of which are 8 bits, and counter / timer 1, which is 16 bits wide.

By default, the clock source for the counter / timers is the MCU clock. The clock frequency may be divided by a number of pre-scale values before being applied to each timer. In this project, no pre-scalar will be used for creating the emitter signal, and a pre-scale of divide by 256 will be used for evaluating the sensor output. No pre-scale gives us the highest possible resolution for the emitter modulation frequency. Any mismatch between the emitter frequency and the internally fused sensor band-pass frequency will result in sensitivity loss.

The timers on the AT Mega have an extensive number of programmable counting modes, which will not be covered in detail in this application note. Conceptually, there are two basic ways to count with the counters: count up from zero to some maximum value and then reset and start over, or count up from zero to some maximum value and then from that value back down to zero. The second approach, called a dual slope method, has a decisive advantage as a way of generating the emitter signal since it allows the frequency and the duty cycle of that signal to be set independent of one another. The TSSP sensors do not require any particular duty cycle, and typically any value between about 20 % and 50 % should work fine. Reducing the duty cycle of the emitter may be used either to reduce the power consumption, or, in conjunction with an increase in the emitter's I_F, to increase the range without changing power consumption.

THE EMITTER

Using no pre-scalar and the Nano's system clock, the timer's resolution is $\frac{1}{16\ \text{MHz}} = 62.5\ \text{ns}$. Using the dual slope method to generate the emitter signal, and given that both slopes are identical in length, it follows that each slope is equal to T/2 of the frequency we want to generate, or 1/(2 x 38 kHz). It is now a simple matter to find out how many counts of the system clock are required for each slope, $\left(\frac{1}{2 \times 38\ \text{kHz}}\right) / \left(\frac{1}{16\ \text{MHz}}\right) = \frac{16\ \text{MHz}}{2 \times 38\ \text{kHz}} = 210.526$. A digital timer can only count integer values, so this value will be truncated to 210. So the timer, clocked at 16 MHz, must count up to 210, then back down to zero for every cycle of the 38 kHz signal. The result of our truncation is a frequency error of 0.25 %, which is negligible.

Counting to 210 can be done with an 8-bit timer, and counter / timer 2 was selected for generating the emitter signal. Each of the AT Mega's timers has two timer / counter configuration registers, TCCRnA and TCCRnB, where n stands for counter / timer 0, 1, or 2. So TCCR2A and TCCR2B are the full names for the counter / timer 2 configuration registers. These two configuration registers have configurable bits, which control:

1. the output pins' behavior,
2. the waveform generation mode,
3. the clock pre-scale value, and
4. an option to force an output compare.

Each of the AT Mega's timers also has two output compare registers, OCRnA and OCRnB (where n again means 0, 1, or 2), that generally define some action to take when the counter / timer matches one of their values. One of these compare registers, OCR2A, will be used to define the maximum count (210) of the timer, and the other, OCR2B, will be used to control when the output pin switches, giving us control over the emitter's duty cycle via the OCR2B value. Each timer / counter also has an interrupt mask register, TIMSKn. The overflow interrupt of counter / timer 2 will be used to count each cycle of the emitter signal, giving us fine-grain control over the emitter burst length. Please consult the AT Mega datasheet for full details on the counter / timers.

The Arduino IDE software generally recognizes the names used in Atmel's datasheets for configuration registers, programmable bits, compare registers, output pin names, and the like. While the newcomer has some work cut out to learn these names and what they mean, the reward for this is an ability to configure the hardware with surprising little, but understandable code.

Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

The complete details for setting up counter / timer 2 to control the emitter signal follows:

TCCR2A
Waveform generation mode 5: WGM20 = 1, WGM21 = 0
Output pin OC2A not used: COM2A0 = 0, COM2A1 = 0
Output pin OC2B disconnected: COM2B0 = 0, COM2B1 = 0 (between bursts) or ...
Output pin OC2B clear on match counting up, set on match counting down: COM2B0 = 0, COM2B1 = 1 (during bursts)

TCCR2B
System clock with no pre-scalar: CS20 = 1, CS21 = 0, CS22 = 0
Waveform generation mode 5: WGM22 = 1
Force output compare not used: FOC2A = 0, FOC2B = 0

OCR2A
Half the period of the emitter signal: OCR2A = 210

OCR2B
Duty cycle of the emitter signal: OCR2B = 63 (for 30 %)

TIMSK2
Enable overflow interrupt, output compare interrupts not used: TOIE2 = 1, OCIE2A = 0, OCIE2B = 0

In C, all these settings can be reduced to the following compact code:

```
TCCR2A = (1 << WGM20) | (1 << COM2B1);
TCCR2B = (1 << WGM22) | (1 << CS20);
OCR2A = 210;
OCR2B = 63;
TIMSK2 = (1 << TOIE2);
```

The interrupt handler for Timer2's overflow event keeps track of how many times it has been called since the start of the burst via a static variable. Once the count reaches the calculated number of cycles in the burst, the output pin OC2B is disconnected from the Timer2, stopping the burst but not the Timer. When the count reaches the calculated number of cycles for the burst period (burst plus pause), the output pin is reconnected to the timer and the counter is reset, thereby restarting the cycle.

THE SENSOR

The time reference of importance for evaluating the sensor output is the period of the 38 kHz carrier signal. Resolving time intervals shorter than one carrier cycle is pointless since that would exceed the accuracy of the sensor. One cycle of the 38 kHz carrier signal was defined in the emitter section as $420 \times \frac{1}{16\ \text{MHz}} = 26.3\ \mu\text{s}$. For measuring long periods like this, it is sensible to use the clock pre-scalar function in the timer. A reasonable pre-scale value would be smaller than 420 in order to resolve the sensor output with sufficient accuracy, such as 256. An 8-bit timer with a pre-scale of 256 will overflow after $256 \times \frac{420}{16\ \text{MHz}} = 6.72\ \text{ms}$. This is not long enough to time the bursts we will be generating, which can be as long as 120 ms to 150 ms. One solution would be to let the 8-bit counter overflow, and count the overflows. A simpler solution from the software perspective is to use the Nano's already available 16-bit Timer1, which gives us a maximum timing capability up to $65\,536 \times \frac{420}{16\ \text{MHz}} = 1.72\ \text{s}$. Another feature available in Timer1 is a noise canceler block that effectively suppresses glitches shorter than $\frac{4}{16\ \text{MHz}}$ (or 4 cycles of the master clock). The TSSP sensors are very high-gain devices, measuring signals that can be barely above the internal circuit noise. This high gain makes them susceptible to glitches caused by disturbances from EMI or RF in the surrounding environment. In addition to the noise canceler, we will use a software filter to eliminate pulses less than 5 ms. There is no hysteresis in the sensor to eliminate the slight instability, which can occur just at the point where the reflected signal is suppressed. The software filter simply ignores any such short pulses. While this does limit the maximum range of the sensor, output pulses less than 5 ms are not that interesting anyway, since they correspond to objects at the farthest possible range limit, where the accuracy of the sensor is at its minimum.

The technique for evaluating the sensor pulse is simple. We use the edge-triggered interrupt available on the Timer1 input pin, D8. When the emitter initiates a new burst, the input unit of Timer1 is set to trigger an interrupt on the next falling edge of the sensor. The interrupt service routine simply resets Timer1, and then sets the interrupt control to trigger on the next rising edge of the sensor. When the rising edge interrupt occurs, the Timer1 value, corresponding to the sensor pulse width, is stored, and the interrupt control is again set for the next falling edge in preparation for the next burst. The stored value is sent to the serial monitor by the main program.

Vishay's TSSP-AGC P Sensor Series for Proximity Sensing

Here are the details for setting up counter / timer 1 to evaluate the sensor output:

TCCR1A
Waveform generation mode 0: WGM10 = 0, WGM11 = 0
Normal port operation: COM1A0 = 0, COM1A1 = 0, COM1B0 = 0, COM1B1 = 0

TCCR1B
System clock with 256 pre-scale: CS10 = 0, CS11 = 0, CS12 = 1
Waveform generation mode 0: WGM12 = 0, WGM13 = 0
Initially, falling edge triggers an interrupt (at start of burst): ICES1 = 0 or ...
Rising edge triggers an interrupt (once a falling edge was detected): ICES1 = 1
Activate input capture noise canceller: ICNC1 = 1

TCCR1C
Force output compare not used: FOC1A = 0, FOC1B = 0

DDRB
Nano pin D8 (AVR PB0) as input: DDB0 = 0

PORTB
Connect pull-up resistor in case sensor is disconnected: PORTB0 = 1

Again in C, all these settings reduce to:

```
TCCR1A = 0;
TCCR1B = (1 << ICNC1) | (0 << ICES1) | (1 << CS12); // falling edge interrupt
DDRB &= ~ (1 << DDB0);
PORTB |= (1 << PORTB0);
```