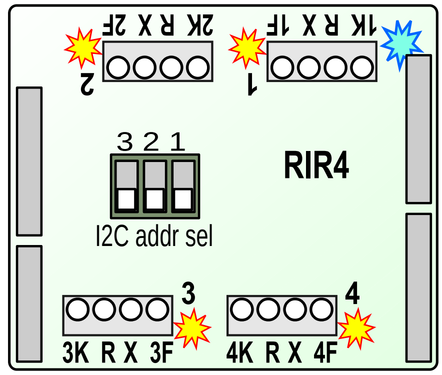


What it is: The RIR4 has four infrared (IR) proximity detectors assembled on a circuit board with connectors that allow it to be plugged into an Arduino Uno, Arduino Mega, or compatible microcontroller.

These detectors allow your microcontroller to sense nearby objects.

The IR sensors have their own light source, so they can be used in total darkness. The RIR4 circuit modulates the IR light of the sensors to eliminate the effects of changing room lighting. The IR light is invisible to human eyes.

Communication between the microcontroller and RIR4 takes place over the two-wire interface (TWI, also known as I2C™). The RIR4 has switches to select one of eight I2C addresses, which allows multiple RIR4 circuits to be connected ("stacked") to a single microcontroller.

Pins used: A4 & A5 for I2C communication, +5v and ground. All other pins are pass-through only.

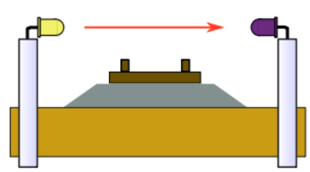
If your RIR4 was supplied with a set of IR sensors, each sensor is equipped with a plastic mounting tube. The tubes are for protection of the sensor leads and to provide mounting support. They are not essential for detector operation and may be shortened or removed entirely to best fit your situation. Just use caution to avoid damaging the leads.
The sensors and circuit are designed for indoor use.

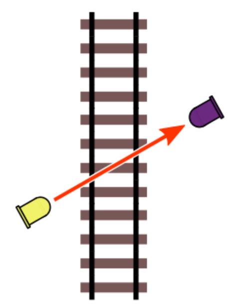
These instructions are tailored for use on a model railway, though the RIR4 may be used for robots and other automation applications.

Installation

Install the sensors: Each sensor pair may be installed in one of two different ways - 'Across Track' or 'Reflective.'

Across Track sensing: The IR LED is positioned horizontally on one side of the track, and the IR photo receiver is placed on the opposite side. A train is detected when it blocks the light path between the LED and photo receiver. The distance between the LED and photo receiver can be up to 18 in. (46cm), or more with careful alignment. Placing the sensors at an angle across the track avoids possible detector flickering caused by the gaps between cars.

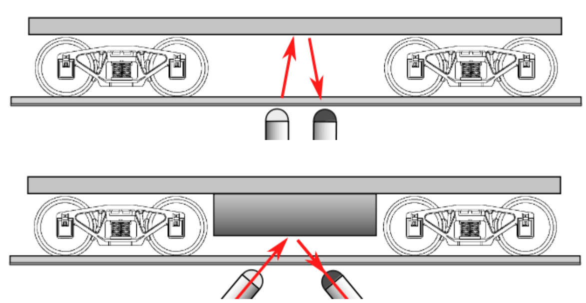




Tip #1 - If mounting the sensors vertically as shown here, slide the plastic tubes away from the sensor then carefully bend the leads to a right angle. The leads are somewhat brittle, bending them more than two or three times may cause a break.

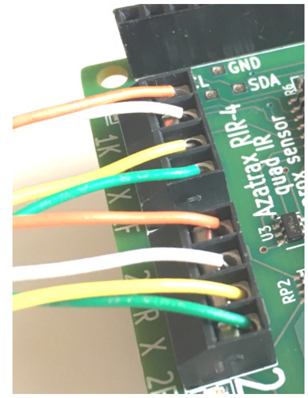
Tip #2 - Locate the photo receiver so it faces away from bright lights or sunny windows. Use scenery or structures to conceal the sensors and shade them from room lighting.

Reflective sensing: Trains are detected when light from the IR LED is reflected off a train and sensed by the IR photo receiver. Typically the sensors are mounted in two 3/16-inch (4.8mm) holes drilled in the roadbed as shown here. Vertical installation works for S and larger scales as long as there is no structure above the track such as a bridge.



For N and HO scale and where the object to be sensed is close to the IR LED and receiver, mount the IR LED and its receiver so they are both pointing at the same spot on the object. Angle the IR LED and photo receiver so their centerlines intersect at the height of the bottom of your rolling stock.

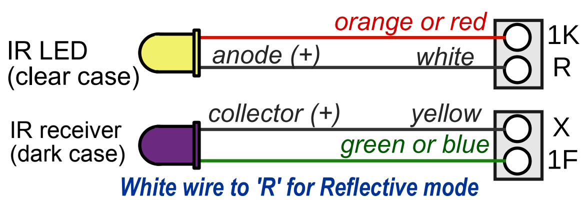
Tip #3 - You can ballast your track after sensors are installed. Cover each sensor with a bit of transparent tape. Apply ballast. When the glue has dried use a dental pick or similar tool to remove ballast from the sensors. An opening of just 1 or 2 mm is needed.

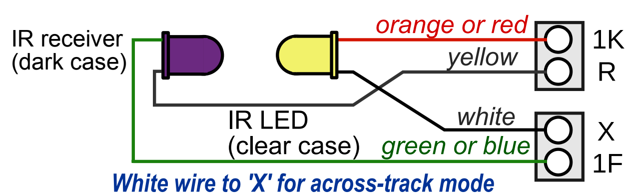


Connecting sensor wires to the terminal blocks:

- ♣ Strip 0.2 inch (5 mm) of insulation off the end of the wire.
- ♣ If the stripped length is too long, the exposed wire may contact another shield plugged on top of this one - this could cause damage.
- ♣ Loosen the terminal screw but do not remove it.
- ♣ Fully insert the wire end in the terminal block top opening.
- ♣ Tighten the terminal screw. Finger tight is sufficient, these are small screws.
- ♣ Bend the wire approximately 90 degrees if another shield is to be plugged on top of this one.

Connect sensor pair #1: Connect the orange or red wire from the IR LED to terminal 1K. Connect the green or blue wire from the IR photo receiver to terminal 1F. How you connect the white and yellow wires to the RIR4 will determine whether this detector will operate in 'Across Track' or 'Reflective' mode. See the diagrams below:





Connect the remaining sensor pairs: The orange or red wire from the IR LED connects to terminal K. Connect the green or blue wire from the IR receiver to terminal F. For all sensors, how you connect the white and yellow wires will determine whether each detector will operate in 'Across the Track' or 'Reflective' mode. Each detector may operate in the same mode or in different modes.

➽ **Pairing is important!** The IR LED that is connected to 1K must be paired on the layout with the IR receiver that is connected to 1F. The same is true for all other sensor pairs.

Power: The RIR4 is powered by the microcontroller's +5v dc. The blue LED on the RIR4 will light when power is on and the RIR4 is functioning.

> <u>On Board LEDs</u>
> The blue LED is on when the RIR4 has power and has completed its start-up sequence.
> The blue LED will 'wink' when the RIR4 receives an I2C command.
>
> Each detector has a yellow LED that lights when that detector is sensing an object.

Test and adjust the sensors:

With no objects near any sensor pair, all yellow LEDs on the RIR4 circuit should be off. If any yellow LED is on, correct the false sensing condition.

To fix false sensing for *Across-Track* mode:

1. Verify that the sensor pair is wired correctly.
2. Make sure the IR LED and photo receiver are pointed at each other, and nothing is between them.
3. Shade the photo receiver from bright lights, and point it away from windows or other strong light sources.
4. Change the nearby room light from incandescent to a fluorescent or LED bulb if possible.

To fix false sensing for *Reflective* mode:

1. Verify that the sensor pair is wired correctly.
2. Pull the IR LED and photo receiver a bit deeper into the roadbed.
3. Infrared light may be 'leaking' through the roadbed material from the IR LED to the photo receiver. Push a metal shim, such as the tip of a hobby knife blade, vertically into the roadbed between the IR LED and photo receiver.
4. Is there an object above the sensor, such as a bridge, or an upper layout level? Mount the IR LED and photo receiver at a shallower angle, or paint the object flat black. Or use across-track sensing.

Are all detectors off?

Now **test for object detection**. Place a locomotive or car at the **#1** sensor. Yellow #1 LED should light. If the LED does

not light, correct the **#1** sensor pair for a false clear condition.
To fix a false clear indication for *Across-Track* mode:
1. Verify that the sensor pair is wired correctly.
2. Adjust the sensor height so the train is fully blocking the light path from the IR LED to the photo receiver.

To fix a false clear indication for *Reflective* mode:
1. Verify that the sensor pair is wired correctly.
2. Adjust the sensors higher or lower in the roadbed.
3. A bright light source above and to the side of the track may be saturating the IR photo receiver. Try pulling it deeper into the roadbed or create shade with scenery or a structure. Change the nearby light from incandescent to a fluorescent bulb.

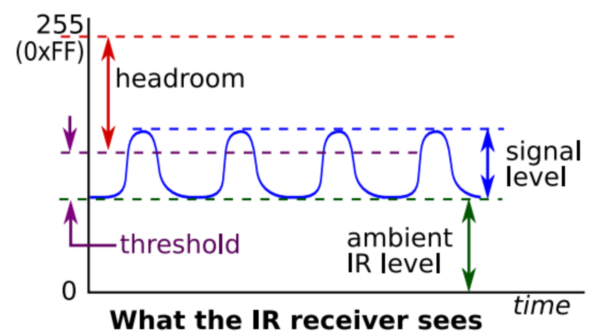
Test with several types of rolling stock and adjust the sensors as needed.

The sensitivity of each IR detector can be adjusted by sending commands via the microcontroller. This is explained in the **I2C Communication** section, page 4.

Detector Details

How the IR detectors work: As mentioned above, each IR sensor has two parts, and IR LED (light source) and an IR receiver. The IR LED continuously sends out pulses of IR light. The receiver is constantly taking in IR light from its surroundings.

The detector circuit converts the intensity of the received IR light into a number. A zero represents absolutely no received IR light. As the intensity of the received light increases, the calculated number increases, up to a maximum number of 255 (0xFF in hexadecimal). When the intensity reaches this maximum number, the IR receiver is said to be "saturated." As the intensity of the received light increases above this saturation level, the calculated number remains at 255.



Detector sensitivity: The detector circuit is looking for variations in the received IR light that match the pulses transmitted by the IR LED. The received pulses ride on top of the ambient IR level. When the height of the received pulses -- the **signal level** -- exceeds the ambient IR level by a certain amount -- the **threshold** -- the detector state changes to "occupied." When the signal level is less than the threshold, the detector state changes to "vacant."

Note that as the ambient IR level increases, the **headroom** on the above chart decreases. Increasing the threshold also decreases the headroom. Headroom is reduced to zero when the ambient IR level plus the threshold equals 255.

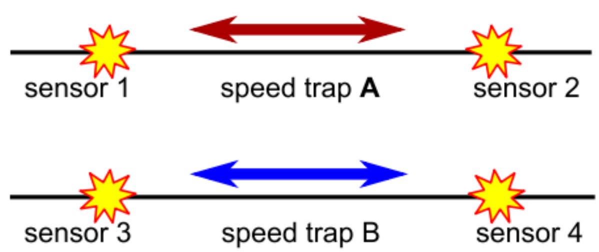
When headroom is reduced to zero, the detector can no longer determine when a signal exceeds the threshold. The detector will never change to the "occupied" state.

Therefor, do not set the threshold to a higher value than necessary, especially in areas of high ambient IR light.

Stopwatches

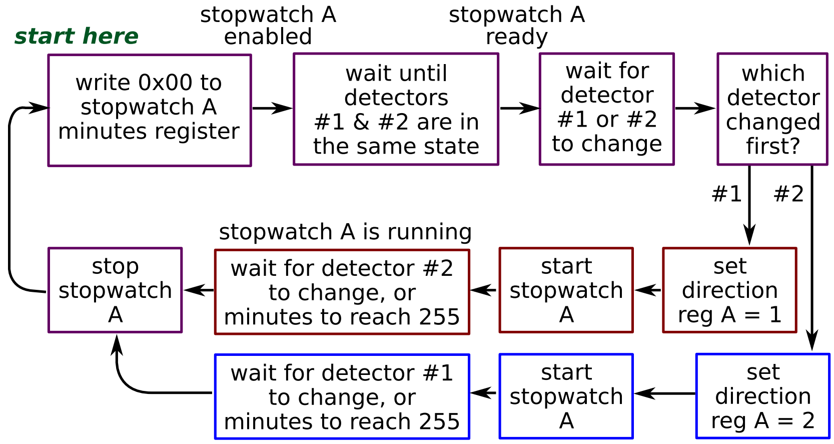
Measuring elapsed time and direction: Two timers -- **stopwatches** -- are embedded in the RIR4 and are controlled by the I2C bus master and the IR detectors. Each stopwatch is individually enabled by the microcontroller.
See **I2C Communication**, page 4.

Stopwatches can be set up as "speed traps," reporting the time it takes an object to go between detector #1 & #2, or between detector #3 & #4. If the distance between sensors in known, the microcontroller can calculate the object's speed.



Stopwatches record minutes, seconds and 1/100 seconds. They will stop at 255 minutes. Accuracy is +/- 2%.

Stopwatch A, after being enabled, is set to "ready" when detectors #1 & #2 are both in the same state, either vacant or occupied. Then if detector #1 changes first, stopwatch #1 starts running. Stopwatch #1 stops when detector #2 changes state.
Direction register A is set to value 01 if the object is moving from sensor #1 to #2. It is set to 02 if the object is moving from sensor #2 to #1.
Stopwatch B, after being enabled, is set to "ready" when detectors #3 & #4 are both in the same state, either vacant or occupied. Then if detector #3 changes first, stopwatch #3 starts running. Stopwatch #3 stops when detector #4 changes state.
Direction register B is set to value 03 if the object is moving from sensor #3 to #4. It is set to 04 if the object is moving from sensor #4 to #3.

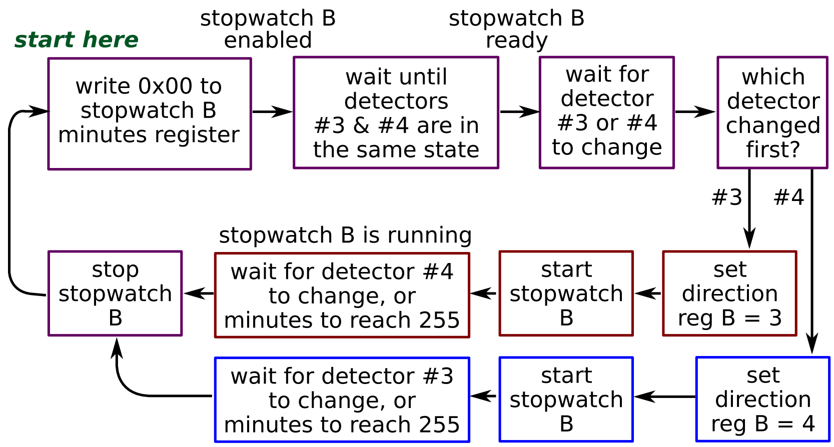


I2C Communication

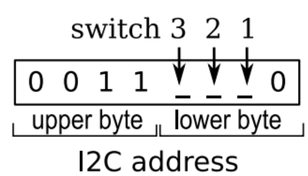
The I2C communication bus is designed to have one 'master' device and one or more 'slave' devices. For the following explanations, the microcontroller (Arduino or compatible) is the I2C master and the RIR4 is a slave.

I2C address: Every slave device must have its own unique I2C address. Three switches on the RIR4 allow you to select one of eight addresses for each RIR4.

Orient the circuit board so that switch 3 is to the left. Push a switch handle **up** to set it to **1**, and **down** to set it to **0**.



Device register addresses: The I2C address switches select the I2C address of the RIR4 (the "device"). The RIR4 has several 8-bit bytes of data ("registers") that the master can read from or write to. Each register has an address. Register addresses are separate from the device address.



switch 3	2	1	address (hex)
0	0	0	0x30
0	0	1	0x32
0	1	0	0x34
0	1	1	0x36
1	0	0	0x38
1	0	1	0x3A
1	1	0	0x3C
1	1	1	0x3E

The I2C protocol specifies that the master initiates each exchange of data.

Read registers: To learn what is happening within the RIR4, the master reads data from the RIR4 read registers. The master first sends the RIR4's I2C device address onto the I2C bus. This gets the attention of the RIR4.

Second, the master sends the register address of the particular data byte of interest.

Third, the RIR4 sends the contents of the specified register onto the I2C bus so the master can read it.

Read Register Contents: This is the information that the master can read from the RIR4 device (addresses are shown in hexadecimal format):

reg. addr.	contents
0x00	Bitmap of the register states. Bit0 (least significant bit) is '0' if detector #1 is vacant (not sensing an object), and '1' if detector #1 is occupied (is sensing an object). Bit1 reports the state of detector #2. Bit2 reports the state of detector #3.

Bit3 reports the state of detector #4.
Bit4 through bit7 are always '0'.
Examples:
If all detectors are vacant, register 0x00 = 0000 0000.
If only detector #1 is occupied, register 0x00 = 0000 0001.
If only detector #3 is occupied, register 0x00 = 0000 0100.
If detectors #4, #3 and #2 are occupied and #1 is vacant, register 0x00 = 00001110.

reg. addr.	contents		
0x10	Detector #1 signal level	0x30	Detector #3 signal level
0x11	Detector #1 threshold value	0x31	Detector #3 threshold value
0x12	Detector #1 ambient IR level	0x32	Detector #3 ambient IR level
0x20	Detector #2 signal level	0x40	Detector #4 signal level
0x21	Detector #2 threshold value	0x41	Detector #4 threshold value
0x22	Detector #2 ambient IR level	0x42	Detector #4 ambient IR level

0x0A Bitmap of the stopwatch states. Bit0 (least significant bit) is '0' if stopwatch A is stopped, '1' if running.
Bit1 reports the state of stopwatch B.
Bit2 through bit7 are always '0'.
Examples:
If all stopwatches are stopped, register 0x0A = 0000 0000.
If only stopwatch A is running, register 0x0A = 0000 0001.
If only stopwatch B is running, register 0x0A = 0000 0010.

0xA0	Stopwatch A minutes	0xB0	Stopwatch B minutes
0xA1	Stopwatch A seconds	0xB1	Stopwatch B seconds
0xA2	Stopwatch A 1/100th seconds	0xB2	Stopwatch B 1/100th seconds
0xA3	Direction A, value 0x01 indicates movement from sensor #1 toward sensor #2. 0x02 indicates movement from sensor #2 to #1.	0xB3	Direction B, value 0x03 indicates movement from sensor #3 toward sensor #4. 0x04 indicates movement from sensor #4 to #3.

Write registers: To control functions within the RIR4, the master writes data into RIR4 registers. The master first sends the RIR4's I2C device address onto the I2C bus. This gets the attention of the RIR4.

Second, the master sends the register address of the particular data byte of interest.

Third, the master sends the contents of the specified register onto the I2C bus so the RIR4 can read it.

Write Register Contents: This is the information that the master can write to the RIR4 device (addresses are shown in hexadecimal format):

reg. addr.	contents		
0x11	Detector #1 threshold value	0x31	Detector #3 threshold value
0x21	Detector #2 threshold value	0x41	Detector #4 threshold value

Notes: ♠ Writing 0x00 or 0xFF to a threshold register will cause it to be reset to the factory default value of 0x18.
♠ Data written to a threshold register will be retained in the RIR4 after power is turned off / on.
♠ After writing to a threshold register, wait at least 4 millisec before sending the next I2C command. This gives the RIR4 time to update its non-volatile memory.

0xA0 Writing 0x00 to this register resets stopwatch A and enables it for the next timing event.
0xB0 Writing 0x00 to this register resets stopwatch B and enables it for the next timing event.

Arduino sketch examples (program code) can be found at **www.azatrax.com/arduino**