# The design of reflected light wall sensors for Micromouse

D.R. Louttit

## Introduction

Micromouse is a set of competitions that have been running since the late 1970s. Some of the competitions are run in a maze made up of vertical white walls set on a black floor. Mice have to run from a designated start position to a designated target position. The layout of the maze is not known before the start.

Mice need to know where they are in the maze, how far away the walls are, and whether particular walls are present.

There are various technologies that can achieve this, but I will only deal with systems where the mouse has a light source and determines the distance to a wall, and its presence, from the reflected light.

You can find more information at: https://ukmars.org/

**Desiderata**

Mice need to be fast and operate in the small clearances of the maze. The walls are either 168 mm apart or 84mm apart depending on the class. This leads to the following desirable features:

1) Small. The mouse will typically be about half the wall spacing in width and length. There is not a lot of space.
2) Light. In general, light mice will be able to turn and accelerate faster. They will need less motor power which also saves weight. They can use smaller batteries, again saving weight. Consider 10 grams/sensor channel as about the maximum.
3) Low inertia. The mouse must be able to change its rate of turn rapidly. If the sensors can be mounted close to the centre of the mouse, it will reduce the moment of inertia.
4) Fast. A competitive mouse can be moving at several metres per second. It needs to be able to detect 12mm or 6mm wide posts at full speed. It thus needs to sense the walls every 5ms or less. Some builders of high-performance mice are looking to sense the wall at a rate of more than 1000 times per second.
5) Sufficient range. It is desirable to be able to detect walls over a range of at least 2 cells. This enables detection of a dead end in time to decelerate and stop before hitting it. It also enables more sophisticated maze solving by finding walls that are visible in adjacent cells.
6) Reliable. If you add a wall that isn't there to the map of the maze, it can be catastrophic.
7) Linear, or at least monotonic. If the slope of the distance/measurement curve varies, this carries across directly into the gain term (KP) of any position controller used. If the gain crosses a critical point, the controller will go unstable. If the curve is non-monotonic, this will again make the controller unstable when the slope changes direction.
8) Consistent. The position data must be the same regardless of the temperature, battery voltage, mouse position and any other effect.
9) Sufficient resolution. If the sensor only produces distance to the nearest centimetre, you will find it very difficult to position the mouse accurately and this can cause the mouse to hit a wall.
10) Cost-effective. Most mouse builders are hobbyists or educationalists. A mouse could need 6 or more sensors and, if they cost £100 each, they are unlikely to be used.
11) Easily set up. If each sensor takes a month of dedicated effort to develop and tune for a particular mouse, builders will be reluctant to do this for 6 sensors.

## Types of reflected wall sensor.

There are at least 5 types of reflected light wall sensor to consider. Some seem obvious, some less so.

**Cameras.**

These are a fairly obvious choice. After all, that's basically how we navigate.

These tend to be heavy and expensive. You may need more than one unless you use unusual lenses. They may need to operate at something over 200 frames per second and this requires a large amount of processor power. Although adding processor power is arguably cheap and easy nowadays, adding the software to carry out the function isn't. It could take more than a thousand lines of code to deal with the data and it has to meet real-time deadlines. Processor power also implies electrical power. This has implications for the battery system. Electrical power implies heat. In a half-size mouse, the thermal load may be a limiting factor. I have had custom plastic parts in a half-size mouse soften due to heat and cause me to abandon the particular approach.

I will not be commenting further on the use of cameras. It would be a case of choosing a camera solution because it is more challenging (fun) than other approaches.

**Time-of-flight sensors.**

At first glance, these seem ideal. They work by timing a pulse of light over the return journey to the wall. This is a very short time. Light travels at about 300mm per nanosecond, so you need to be able to resolve time down at the 10 picosecond level. Don't expect to do this at home unless you are a talented optoelectronic engineer with an extensive set of expensive test equipment. You will spend a lot of time and money.

There are integrated time-of-flight sensors available at affordable prices. You simply mount them to point in the right direction, send the proper bytes over a serial link and receive the target distance in millimetres over the serial link. They are insensitive to the reflectivity of the wall.

The big issue is that they take at least 10ms to take a reading. It is also not clear exactly which part of the wall is giving the reading. I spent a lot of effort trying to make these work in a reasonably fast mouse and gave up in the end. If you want to have a very slow mouse, these are a very attractive solution but I will not be covering them here.

The device I investigated is the STM VL6180X

**Triangulation sensors.**

These work by shining a spot of light at an angle to the wall and detecting the position of the spot in the field of view of the sensor. These are relatively insensitive to the reflectivity of the wall and can be made immune to ambient light.

The geometry of the system is quite tricky. It requires good optics and precision construction.

The light source is quite straightforward but it needs a narrow beam. A laser would be ideal but these are heavy and comparatively large. Safety must be considered.

The sensor is essentially a one-dimensional camera. The best-known supplier is Hamamatsu Photonics. The sensors can be tricky to get hold of. The relevant page of their website is here:

https://www.hamamatsu.com/eu/en/product/optical-sensors/distance-position-sensor/psd/one-dimensional-psd.html

There have been successful mice using this technology so it may be worth further investigation. I have little experience of this method as I was put off by the optics. It can be tricky to get sufficient signal to noise ratio.

I will not be covering this method further.

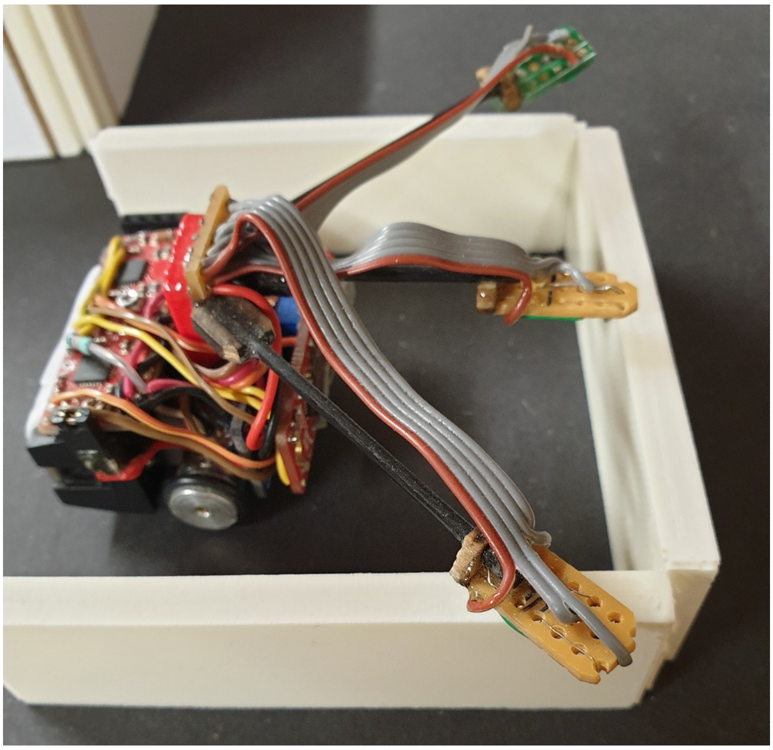
**Top-of-the-wall sensors.**

This essentially consists of a line-follower sensor looking down at the top of the wall. It is cheap, well understood and has been used in successful mice in the past. It can be made fast, insensitive to the wall reflectivity, and immune to ambient light.

The significant issue is that the sensors have to be mounted a long way from the centre of the mouse. This leads to a large moment of inertia. Most people choose not to use this method as they have ambitions of achieving a fast mouse and the large moment of inertia puts them off.

The moment of inertia is proportional to the square of the offset. Consequently, it is reduced to a quarter of the value when used on a half-size mouse compared to a full-size one.

I think it would be worth considering for a modern half-size mouse as the weight of surface-mount components is greatly reduced compared to the components used in the past. Carbon fibre will allow very light construction. However, I will not cover it here as the subject of line-following sensors has been covered elsewhere.



**Reflected light amplitude sensors.**

This is the dominant technology and many high-performance mice use them. I shall deal with the design considerations in considerable detail.

These work by measuring the amount of light reflected off the wall. Clearly, the brightness of a light diminishes as you get further away.

In order to cancel ambient light, these sensors need to make one measurement with the light source on and another with the light source off. The “off” reading is the ambient. The final value is the “on” value minus the “off” value. Provided the time interval between the readings is small, the ambient light can be cancelled effectively.

They can be fast, cheap, light, and mounted close to the centre of the mouse. The software is relatively simple.

There are drawbacks. They are affected by the reflectivity of the wall. The transfer characteristic is intrinsically non-linear. Long range can be tricky to achieve.

## Optical devices and geometry.

**Light sources**

Any light source must be capable of switching on and off rapidly. The reasons for this will become clear later.

LEDs and lasers are the only viable technologies for light sources. The important parameters are:

Angle of half intensity/beam shape

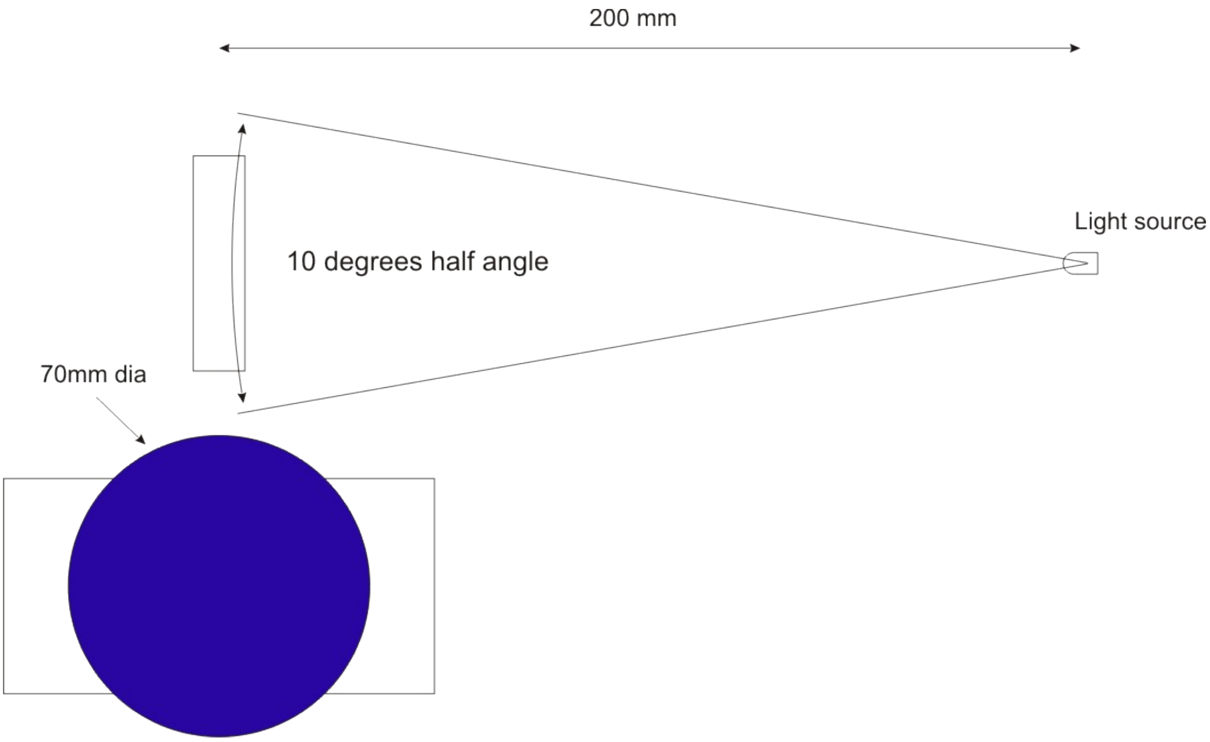
Maximum power output

Wavelength

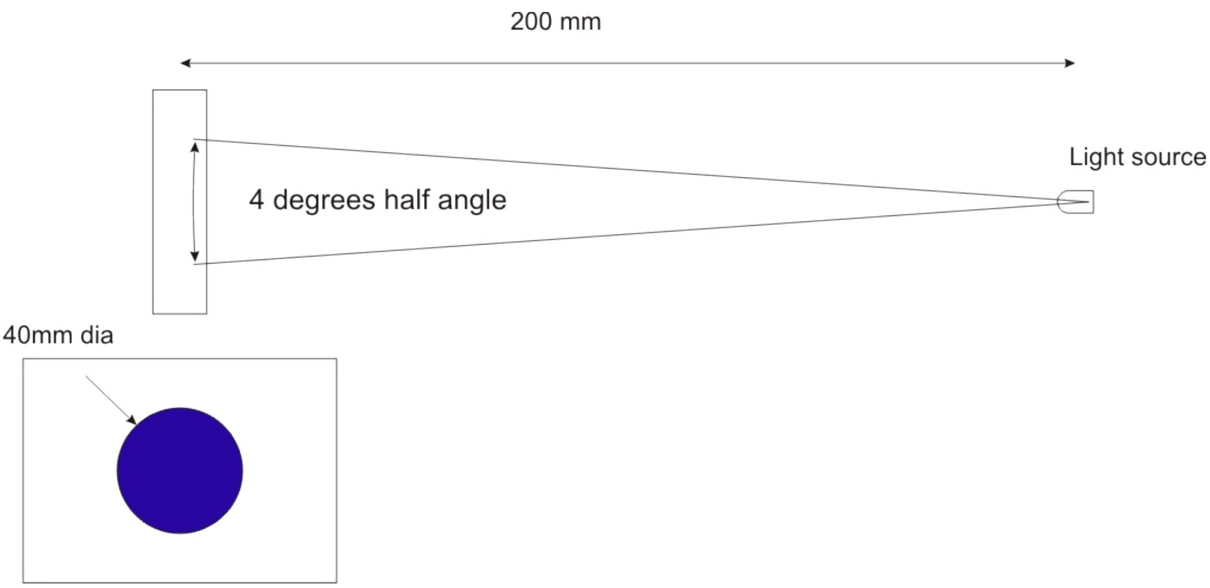
Size and weight

Cost

**Angle of half intensity.**




You can see that a light source with a 10 degree half angle (20 degree included angle) will spill some light over the top of the wall and down onto the floor at a range of 200mm.




A 4 degree half angle will not spill over the wall even at 200mm range.

I will come back to this topic later, but, for the moment, narrower angles are better. Lasers have a half angle of pretty much zero, so they are particularly good.

There is at least one visible LED where the beam is not a round spot but is more of a square with a hollow centre. This is probably an image of the LED chip used.

**Maximum power output.**

The bigger the power output, the better the signal to noise/ambient. All things being equal, you will get better range the bigger the power output.

However, there can be issues to do with eye safety especially with a laser. Lasers should be limited to 2mW maximum.

Some of the infra-red LEDs should be considered carefully as the power output can be very significant. The peak power can be very high for a very short time and with a low duty cycle. This is probably safe but you should consider how to ensure that the LED cannot be inadvertently left full on due to software or hardware faults. More on this in the section on driving electronics.

**Wavelength**

The choice is really between red and infra-red.

Infra-red LEDs have lots of advantages and are preferred by many constructors. They are cheap and can easily be bought in beam-widths down to 4 degrees. They are a good spectral match to cheap silicon sensors. The ambient infra-red level from sunshine is less than from visible light.

A disadvantage is that the I.R. beam cannot be seen so it is difficult to see what is going on as the mouse moves about the maze. There may be reflections off multiple walls that you had not considered. There is no easy way to determine if the spot is circular and even.

A possible safety issue is that I.R. cannot be seen and so does not generate a blink or aversion response to prevent damage to the eye.

Of the available colours, red is preferred to others as it is a better match to silicon sensors whilst still being visible.

There is at least one red LED with a beam width of 4 degrees.

UV combines the worst features of the other LED colours available with the need for everyone to wear protective goggles!

White LEDs often use a phosphor to convert blue or UV light to white. These are slow in turning on and off. I wouldn’t recommend them.

At other wavelengths the reflectivity is undefined/unknown. Radio waves, as in radar, will probably not work.

**Size and weight.**

LEDs of all types typically weigh under 0.5 grams. You will need the 5mm diameter type to get the optical characteristics and they are about 8mm long.

Laser modules weigh about 2 grams and are about 7mm diameter and 15mm long.

**Cost**

Suitable LEDs can cost between about £0.1 and £1 each. The better the specification, the more you typically pay.

Laser modules can be bought on Ebay for as little as £1 but the sky's the limit on price really.

**Conclusion**

I personally prefer visible light as it has shown up issues for me in the past. I think the majority of people use IR as they seem to be able to get better range with them.

**Sensors**

These should be matched to the light source for spectral response. Don't use something with an IR filter for visible light!

Cadmium sulphide photoconductive cells are not suitable as the response time is of the order of milliseconds and is far too slow. They are also considered a pollution hazard when they are disposed of.

Photodiodes and phototransistors are the only viable options. The important parameters are:

Angle of half sensitivity

Area

Speed

Sensitivity

Linearity

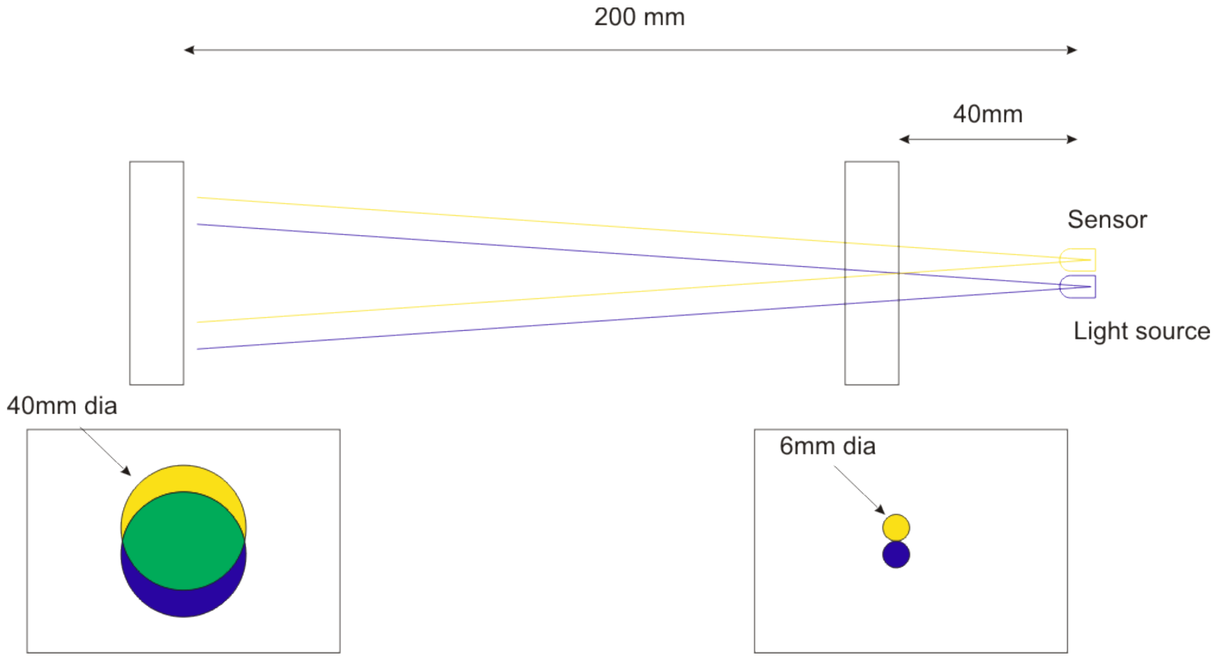
Wavelength and filtering

Size, shape and weight

Cost

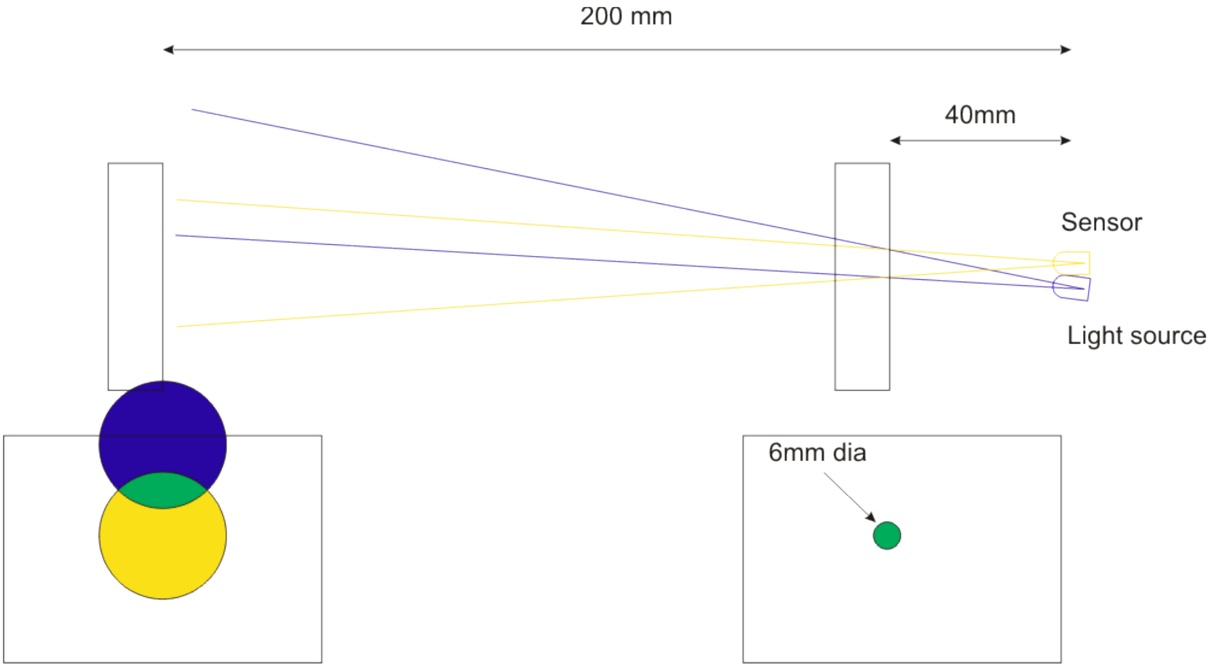
**Angle of half sensitivity**

I think of this as the field of view of the sensor. Anything outside the field of view is invisible! (Not quite true)

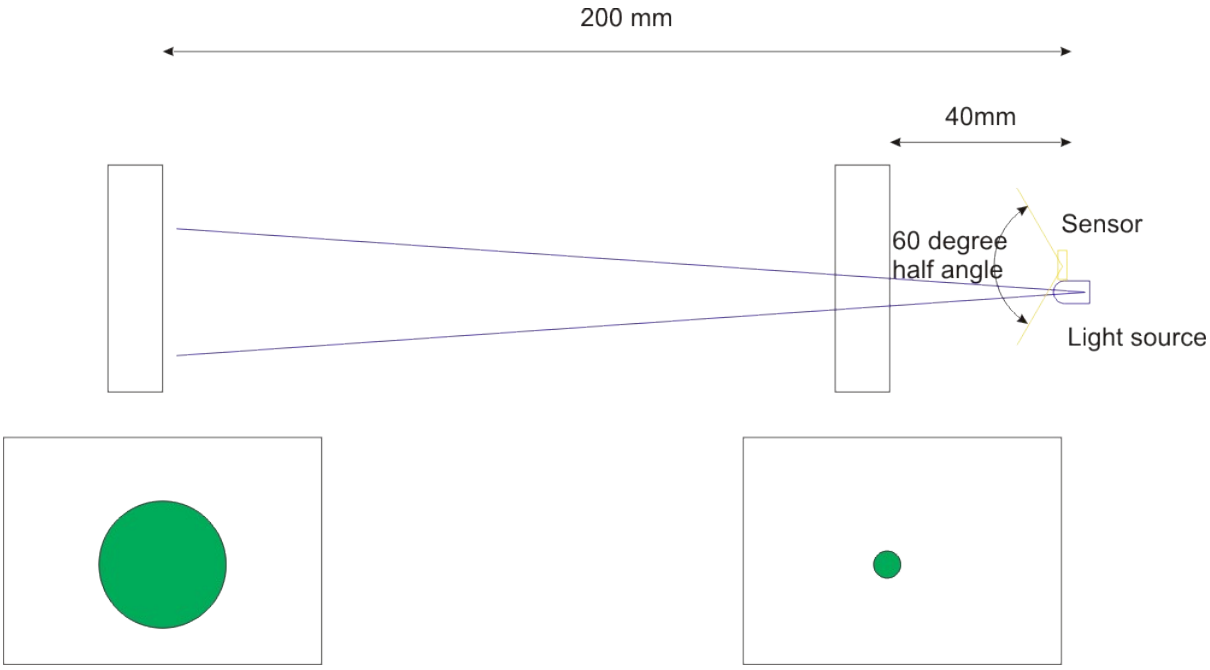



The diagram shows the field of view of a 4 degree sensor alongside the illumination from a 4 degree LED. The sensor and LED are as close as they can realistically be; a 1mm separation between two 5mm diameter parts.

You can see that the sensor can see most of the illumination of the LED at 200mm but nothing at all at 40mm. You can angle the sensor/LED pair towards each other to change the distances at which the illumination is visible, but you cannot get all the illumination inside the field of view for all distances. This geometric effect will change the relationship between signal and distance.




If you angle one of the devices to get best coverage at 40mm, you can see that the overlap is reduced at longer distances. There is no good solution with two narrow-angle devices.




Sensors are available with a wide field of view. You can see that a 60 degree half-angle type will have no geometrical issues of this type. The drawback is that it is exposed to a much wider angle of ambient light.

You can use shading blinds to reduce the viewing angle of sensors to any desired amount. If you want something really extreme, you can use mirrors, blinds, and lenses to achieve any desired field of view.

**Area**

Sensors can have tiny areas of silicon or up to something about 5mm square. The bigger ones collect more light and give bigger outputs. They cost more.

Large sensors are more tolerant of alignment errors when you try and do complex optics.

A tiny sensor with a big lens to collect light will have a small field of view.

**Speed**

Faster is better.

A fast mouse could be moving at several metres per second. It will pass a 12mm post in a few milliseconds. You might want two or more readings while this is happening. This is why cadmium sulphide photocells aren’t suitable.

As we will see when I discuss electronics, we want the whole sensing process to be as fast as possible.

Phototransistors have settling times of the order of 10s of microseconds. Photodiodes settle in the order of 1 microsecond.

**Sensitivity**

Think in terms of electrons/photon. Photodiodes will give a little less than one electron/photon. Bigger areas collect more photons and are more sensitive.

Phototransistors are exactly equivalent to a photodiode feeding into the base of a transistor. Whatever the photodiode current is, it is multiplied by the gain of the transistor. Expect currents about 100 times greater.

**Linearity**

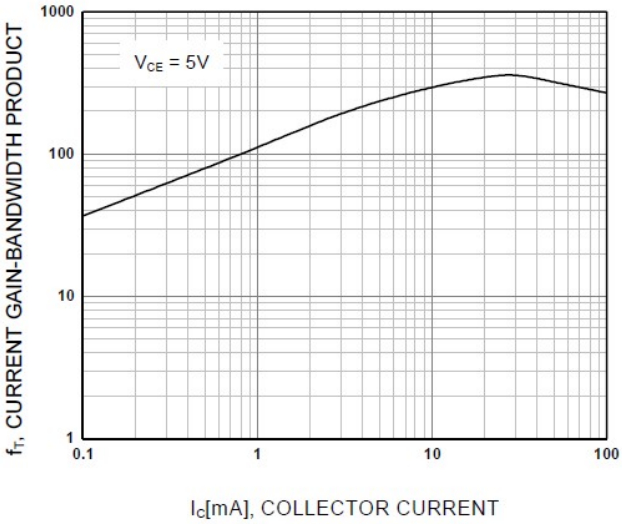
We care about linearity because ambient light can be much larger than the desired signal and also can vary over a wide range between looking directly at the light and directly away from it. As the mouse moves, this is almost inevitable.

A photodiode is almost intrinsically linear. The slope will be the same in terms of photons/electron however many photons are passing.

Phototransistors are less linear. The gain of the transistor varies with the bias current. At low currents the gain reduces and at high currents the gain reduces again.

With a photodiode, a certain change in light level might give a 1uA change in output current in the dark, the same change in output current in a lighted room, and the same again in full sunlight when the total photocurrent might be 100uA or more.

The same change in light level might give you 70uA change for a phototransistor in the dark, 100uA in a lighted room and back to 70ua again in sunlight when the total photocurrent might be 10mA or more.




I only use photodiodes for wall sensing. Phototransistors are fine for line following when the sensor can be well shielded from ambient light.

**Wavelength and filtering**

Silicon is most sensitive in the infra-red but many types give good response with red light.

Many types are avaliable with infra-red filtering. These are much less sensitive to ambient sunlight where the majority of the energy is in the visible range.

Historically, strong artificial light had very large infra-red content and caused many issues. Fortunately, modern LED lighting generates far less infra-red.

**Size, shape and weight**

Photodiodes and phototransistors can be as small as 0603 smd and photodiodes as large as 40mm diameter. Bigger ones are inevitably slower. The 5mm cylindrical through-hole types tend to have a comparatively narrow field of view of 20 degrees or less. Wide field of view types tend to be in surface mount packages. Weight is basically proportional to size and the 5mm cylindrical types are about 0.5 grams.

**Cost**

The cost can be of the order of £0.10 for the smallest, surface mount, types. Cylindrical types come in under £1 and the biggest ones can cost more than £100!

I use surface mount visible types with the largest area I can afford at under £1 each.

## Electronics

### Driving the LED/laser

The LED needs to be driven with a high peak current to generate the maximum possible light. Most LEDs can tolerate a peak current of over 1A for a very short period.

This current must be isolated from the main supply to avoid making unnecessary supply ripple that could cause issues with the sensitive parts of the mouse. In extremis, it could cause a brownout reset for the microcontroller with unfortunate results.

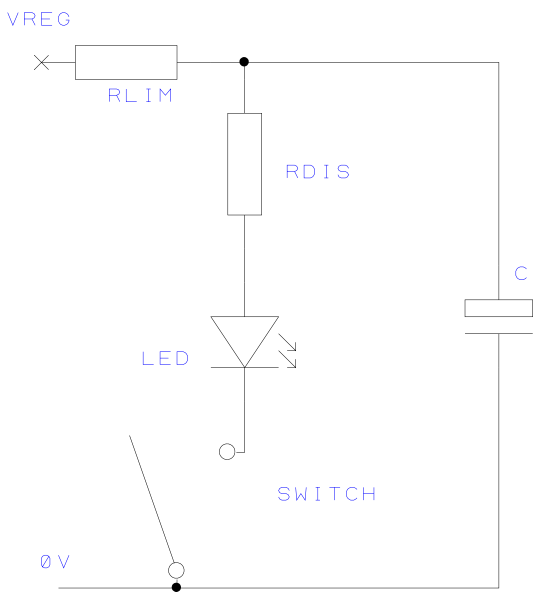
The current must be sustained for long enough to make the light measurement. Faster sensors and ADCs enable to current pulse to be shorter.

Current pulses must be of uniform amplitude so that the LED light is the same for each sample.

The overall duty cycle must be low enough so that the LED is within its thermal limits. Small duty cycles will also minimise supply current.

So, if the LED can tolerate, say, 250 mw, continuous power and has a forward voltage of 2.0 volts, the maximum average current is 125 mA. This makes the maximum duty cycle for a 1A drive current 1:7. If you want to sample every 0.5mS, the maximum “on” time for the LED is 42us or so. Some external ADCs with slow driver software will take longer than this so be careful.

In simplified form, classical LED drivers look like this:




The switch can be a transistor or FET which is turned on and off under software control. More on this later.

VREG is a regulated voltage within the mouse. If it is not regulated, the light output will vary with the supply voltage. It must be comfortably more than the forward voltage of the LED. This forward voltage is higher for visible LEDs than IR ones but, even for an IR LED, it can be about 2.5V at 1A forward current. Expect variation between LEDs of at least 10% in forward voltage. It is also temperature dependent; reducing at about 2mV/degree centigrade.

RLIM should be selected to limit the average current from the supply to a safe value. Safe could mean "doesn't blow the LED up" or "doesn't permanently damage your eye". For a 5V supply, I typically use 47R for RLIM which limits the current in all cases to about 100 mA and, with a working LED, typically 50 mA. Most LEDs will tolerate this indefinitely. The reason for limiting the average current is that, sooner or later, your software will turn the LED on and then not turn it off at the planned time. It is best if software bugs don't destroy the hardware as we are all prone to putting bugs in our code!

C is charged through RLIM with a time constant of RLIM*C. We want it to be stable at VREG before each pulse of the LED so that the LED current is the same each time. So RLIM*C needs to be much less than the time between readings. You may get away with shorter times between readings, but you could then find that LED current depends on exactly how much time occurred between two pulses.

10uF capacitance with 47R resistance gives a time constant of 470 microseconds, so you may get away with a 10uF capacitor if the time between pulses is 2mS or more. 5mS would be better.

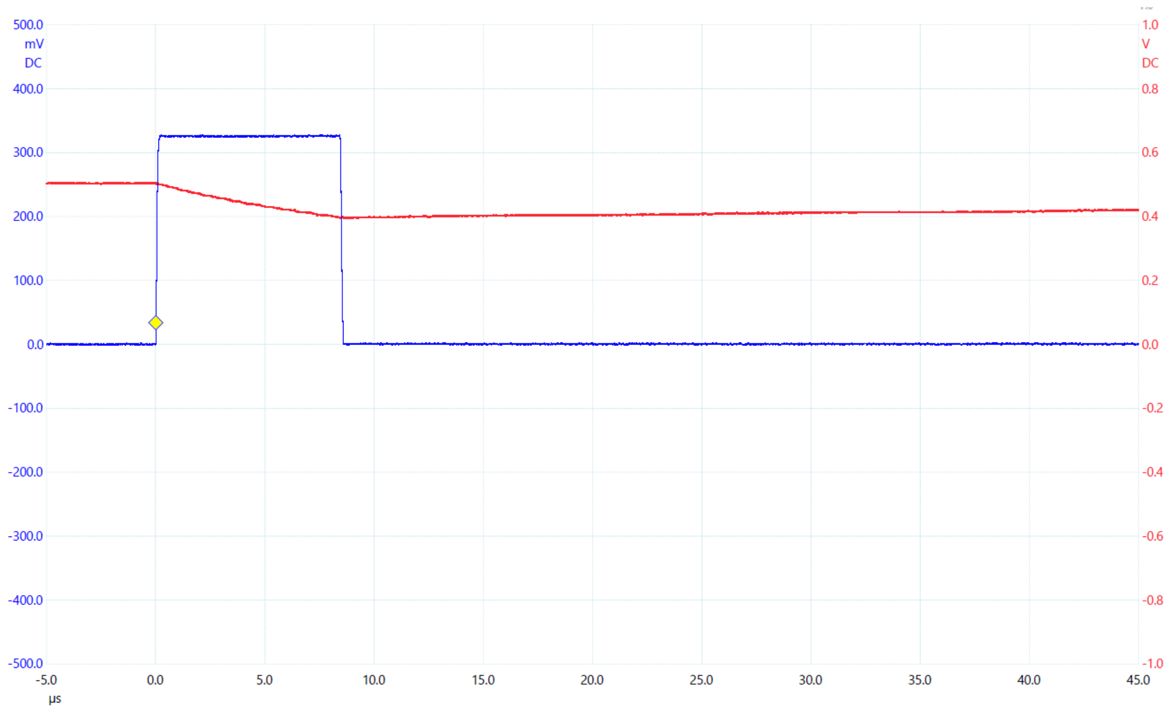
You can reduce RLIM to get a shorter charging time at the risk of your LED or eye. You could decrease the value of C at the cost of energy available to the light pulse.

When the switch closes, C discharges through RDIS and the LED. The peak current should be as high as the LED can tolerate and probably about 1A. The peak current is (VREG-VF)/RDIS. The ESR of the capacitor is included in RDIS. With a 5V supply, RDIS should be about 2R2 for a red LED and perhaps 2R7 for an IR LED.

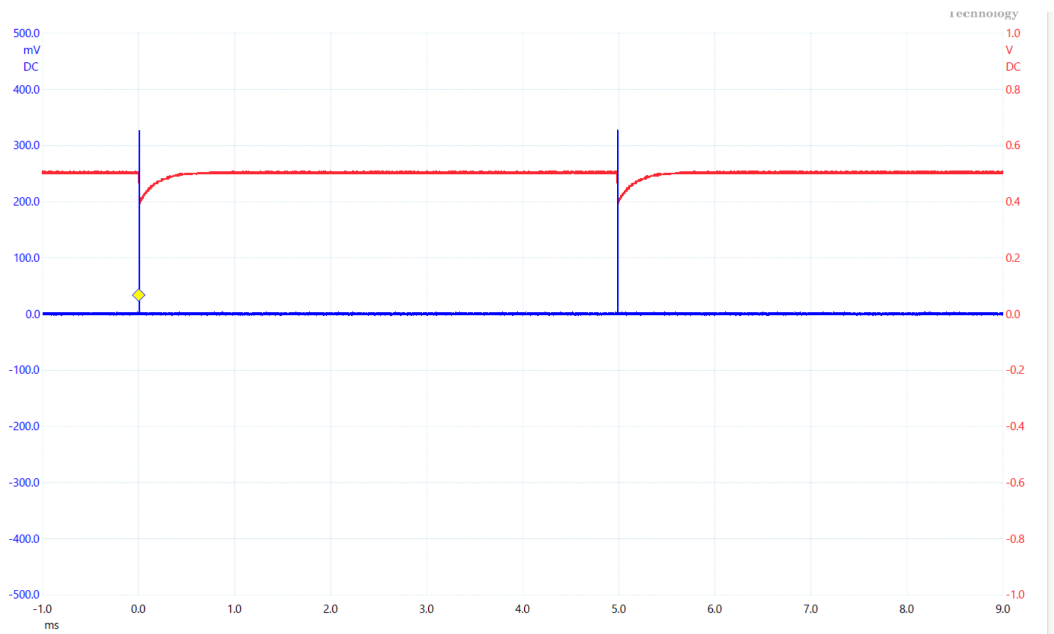
Note that RDIS carries 1A or so of current. I don't trust 0603 resistors at this current level and use a 1206 instead.

Some electrolytic capacitors can have an ESR of more than this so this type is to be avoided. Ceramic types are to be preferred but there are many types of ceramics so choose with care. In general, physically bigger capacitors are better but I have had good enough results with 0805 size.

As C discharges, the voltage across it drops and so does the LED current. LEDs are fast so the light will achieve maximum intensity within about 1microsecond and will then fall. You should aim to make your returned light reading as soon as possible after the switch turns on. The fall in LED current is not exponential but the rate of fall increases as the time constant RDIS*C decreases. This is the cost of a smaller value of C.



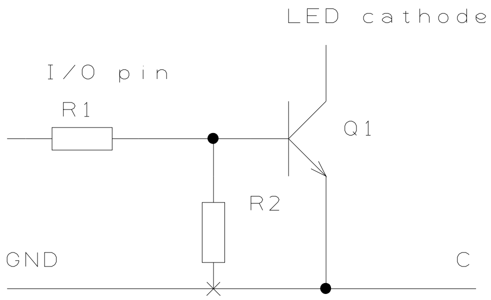

With my current system, the voltage across my 10 uF capacitor drops by nearly a volt during the 8 microsecond LED pulse and then slowly recovers.




The high peak current flows in a loop consisting of C, RDIS, the LED and the switch. You should make this wiring as short as possible using heavy gauge connections. Make just one connection between this loop and the negative supply. If a wire or track within the loop is shared with a sensitive circuit, it can cause errors. In particular, keep the ADC feeds out of this loop.

**The switch**

This is normally a transistor or FET. I will not consider other options here.




Any transistor must be carefully chosen. Don't just pick what is at hand or cheap. Something like a BC547 or BC337 will not do.

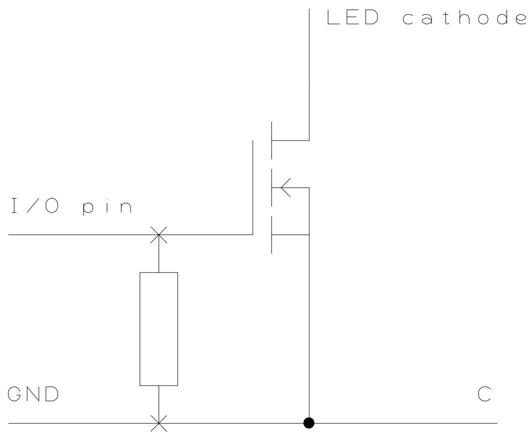
The peak collector current is 1A or more. The collector-emitter saturation voltage of a BC547 is 0.6V at 100mA and more at higher currents. This is subtracted from the voltage available for the LED. 2.7 volts of forward voltage plus 0.6V of loss in the switch means that you can't get 1A through the LED at 3.3V supply.

There are good transistors but they cost more. They still may not be suitable. For example, the ZTX653 has a saturation voltage of only 0.3V at 1A. It needs 100mA of base drive to achieve this and the maximum output current of most microcontrollers is less than 20mA. You could use an extra transistor to provide this base current but this adds cost and complexity.

R1 should be chosen to limit the current from the I/O pin. If the microcontroller can deliver 20mA and is running from 5V, the resistor should be 5/0.02 = 250, say 270R or 220R.

R2 is optional for most modern microcontrollers that can pull the I/O pins down to 0V. If used, 10K is a suitable value. It doesn't actually improve the performance of the circuit and costs weight, space and cost, but not much for an 0603 resistor.

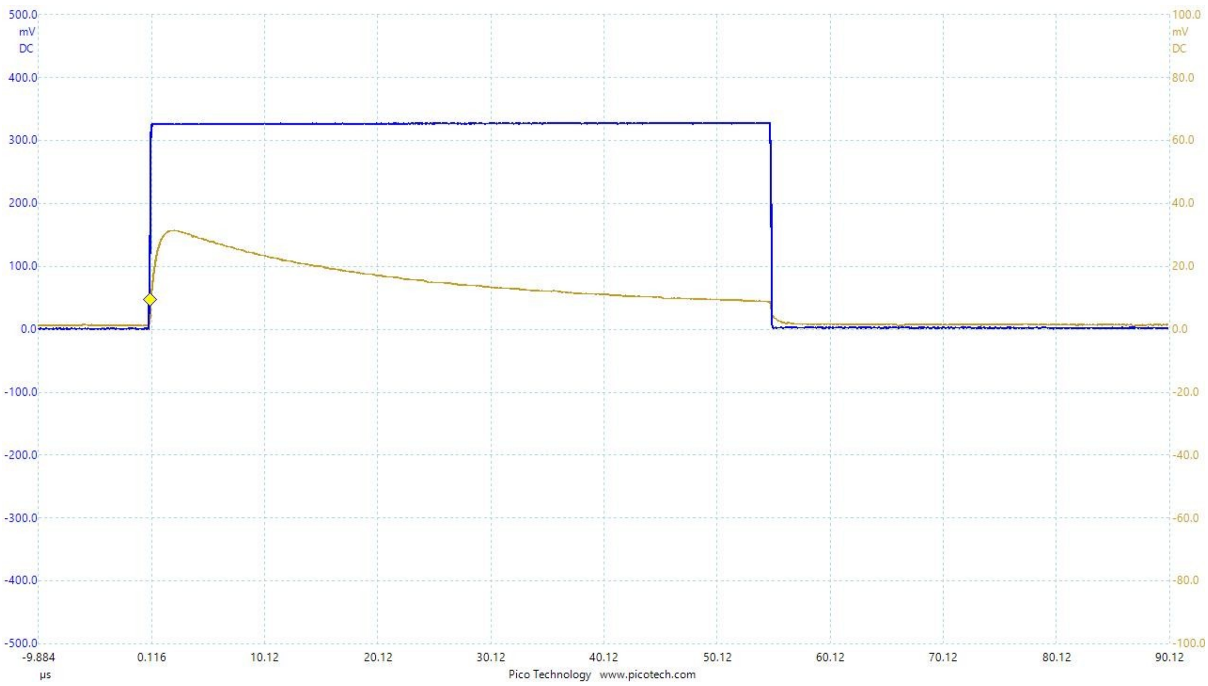
I wouldn't use a transistor.




NFETS are a good solution. You want something with an on resistance of 0.1ohms or less. This will only lose 100mV of the available voltage. You will need a gate threshold voltage of 2.5V or less so that the gate can be driven directly from the microcontroller.

There are many types that will do the job but nearly all are surface mount. If you must have through hole, you are probably looking at a TO220 package costing more than £1 and with weight and space costs. I use SOT23 types at less than 50p.

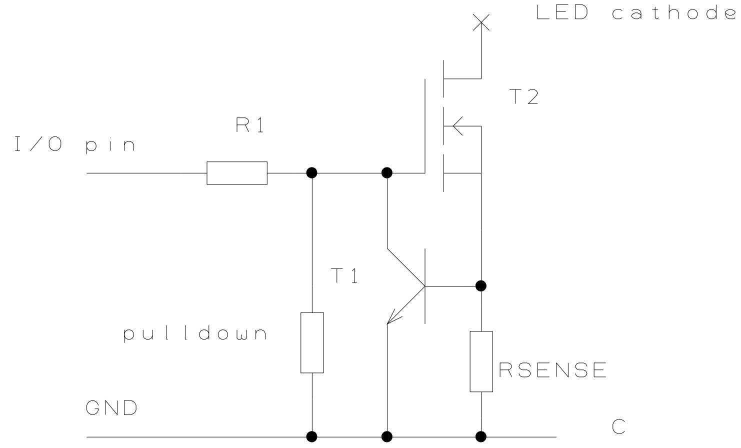
The function of the resistor is to pull down the gate and ensure that the transistor is off during controller reset. If you don’t do this, the gate of the FET will float and the FET may turn on. An extra few hundred mA (dependent on how many LEDs your sensor system has) during a processor reset could cause issues. Resistors are cheap!




This is the actual light waveform from the LED driver. To get the best signal, you need to sample very soon after the LED is turned on.

**Novel circuit**

The existing systems clearly work but they have some disadvantages. They need a regulated voltage supply. 3.3V is marginal, 5V or more is better. The LED current, and hence light, is not particularly well defined. I have come up with what I think is a better solution that is only slightly more complicated:




Rather than start with a lowish regulated voltage and try and control the current with a series resistor, I start with a constant current system.

The FET, T2, is the switch. When the I/O pin goes high it switches on and supplies the LED with current that has to pass through RSENSE. The voltage across RSENSE is applied to the base of T1 and when it reaches 0.6V, T1 will switch on, pulling the gate of T2 lower until the FET turns off just enough to maintain the voltage across RSENSE at that level. Changes in the voltage at the drain of T2 have no effect until the drain is close to the source voltage.

If RSENSE is 0R56, the constant current will be nearly 1A. I would again use a 1206 size here.

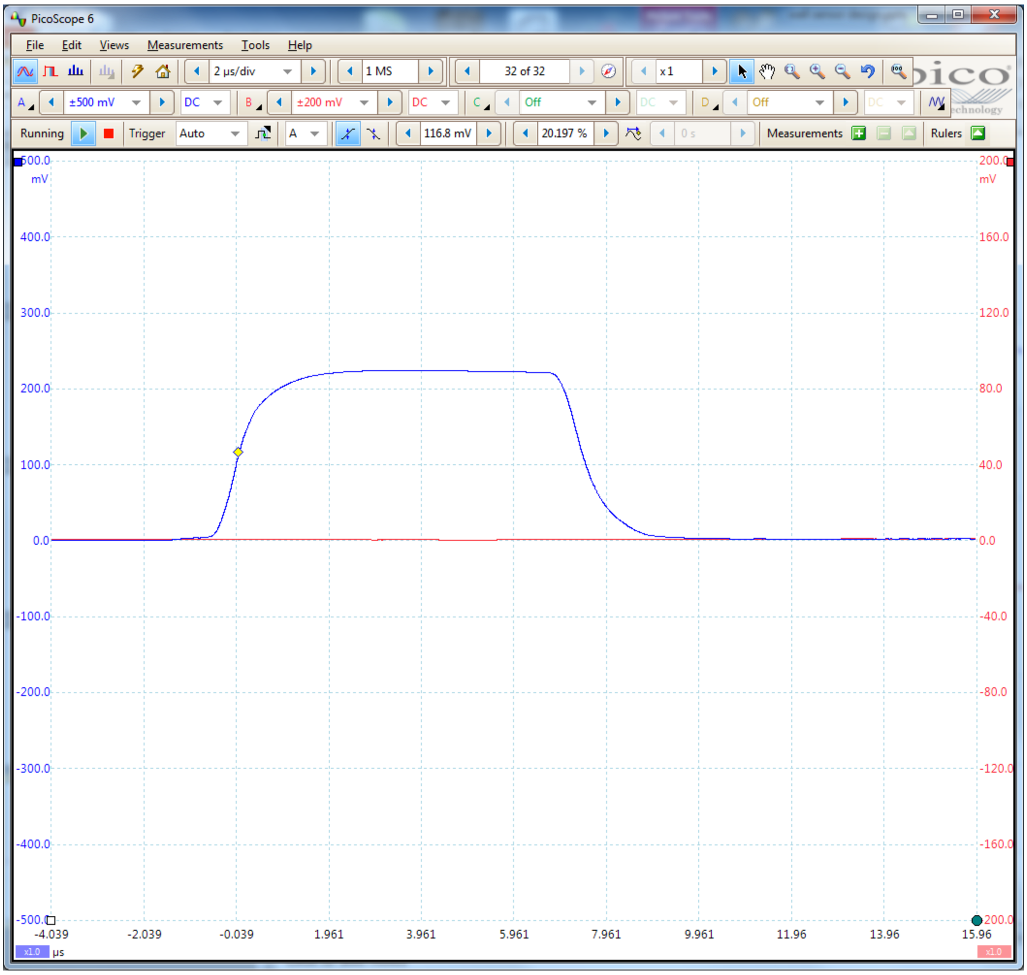
The current source means that C can discharge to a voltage of about 1V more than the forward voltage of the LED without any effect on the light level. Timing accuracy of the ADC sampling point relative to the turn-on timing of the LED is no longer important.

It also means that the circuit can be fed from the battery voltage without any regulation.

Provided that C has charged up a reasonable amount, the next pulse can happen. It means that you can have the time constant RLIM*C bigger and still sample the wall at shorter intervals. Sampling at 0.5mS with 47R and 10uF is feasible without concerning yourself about timing jitter affecting the light pulse level.

R1 can conveniently be 1K. Don't make it too big or it will slow down the rising edge of the light pulse. The pulldown resistor can be 10K.

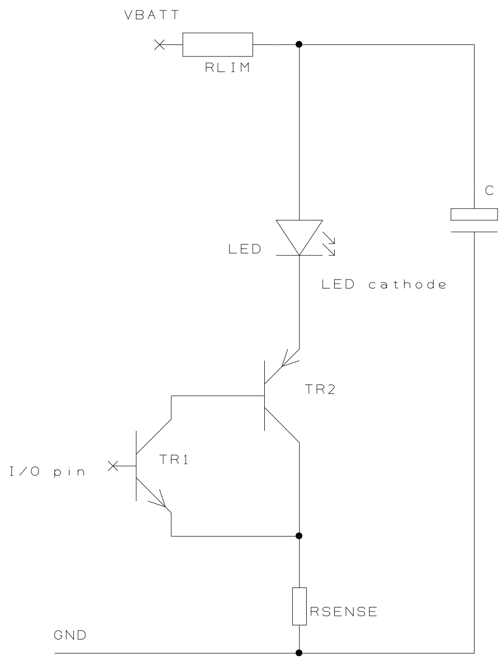
This circuit needs no current limiting resistor for the LED so RLIM can be 0 (a bit of PCB track.




Light output from constant current circuit.

**Through hole version**

When talking about the new circuit, it came out that FETs are difficult to source in small through-hole variants. I thought about it and came up with this for people who need a through-hole solution:




TR1 and TR2 form a complementary Darlington pair. TR2 needs to have a current capability sufficient for the peak LED current. I use the SS8550CBU from RS, part number 739-0347 at about £0.10. TR1 is a cheap NPN type, such as BC548.

The I/O pin is used as a voltage reference. The voltage across RSENSE when the I/O pin is high is the microcontroller supply voltage minus TR1's VBE (about 0.7volts). With a 3.3 volts supply, the voltage across RSENSE is about 2.7 volts. A 2R7 resistor will define a current of 1A. If the current drops, the base emitter voltage for TR1 increases and increases the current. There is thus strong negative feedback to stabilise the current.

The collector current of TR1 is the base current of TR2 and the overall current gain is very large. The current drawn from the I/O pin is less than 1mA.

The system works until the emitter voltage of TR2 drops below about 1 volt more than the I/O pin voltage and then falls into saturation.

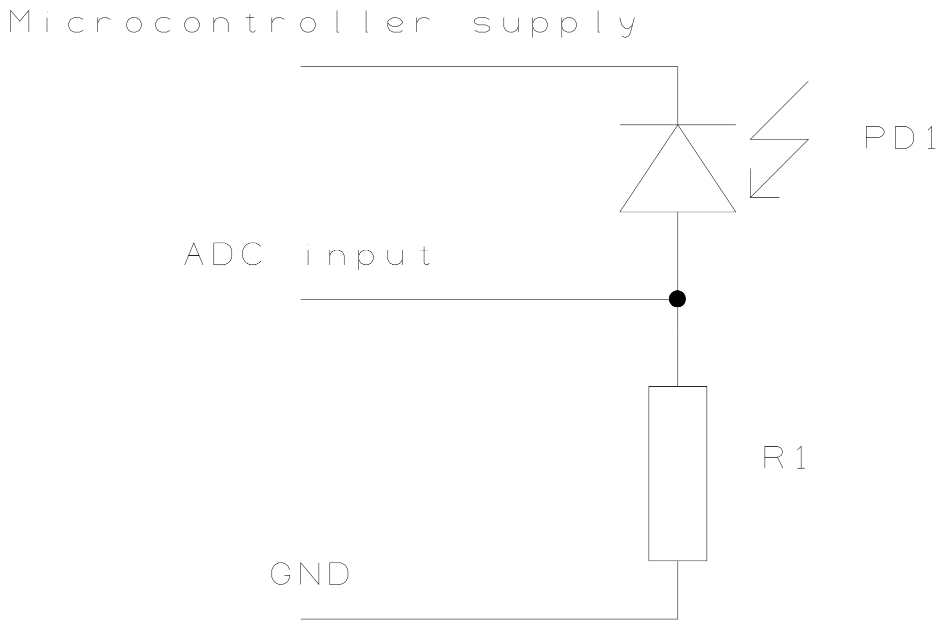
So, for a 3.3 volt microcontroller, the battery voltage must be at least 3.3V + 1V + LED forward voltage + some overhead; say 9 volts. For a 5V microcontroller you will need a higher voltage battery.

Because this is a constant current driver, the voltage across C doesn't have to be stable. You can use a large capacitor, say 100uF, to ensure that there is enough charge for a longer discharge time if needed.

RLIM should be scaled according to the supply voltage. 82R is about right for a 9V supply.

**Photosensor**

This is what I have been using for years:




The photocurrent through PD1 passes through the load resistor R1 and produces a voltage across R1 that is fed directly to the ADC input. The current through PD1 is a tiny leakage current plus the photocurrent. It appears as a constant current device with a high impedance and gives good rejection of supply noise. Swapping R1 and PD1 gives less supply rejection. I normally decouple the supply voltage anyway.

In theory, the maximum voltage that can be supplied to the ADC is the microcontroller supply so that the input will not be overloaded. This may not be true for higher light levels as the photocurrent can be sufficient to forward bias the photodiode, when it acts as a photovoltaic cell. The current into the microcontroller pin is the maximum photocurrent achieved minus the current through R1. The microcontroller will survive any likely light level, but keep away from Electromagnetic Pulses!

Make sure you get the photodiode the right way round or you will get strange results.

A phototransistor could be used in place of the photodiode but make sure that the collector goes to the microcontroller supply. If you use a phototransistor, the lowest voltage possible is the collector saturation voltage at about 0.4V. This will limit the swing into the ADC input.

You choose R1 so that, at the maximum expected light level (brightest ambient plus closest shiniest wall) the ADC just fails to give maximum count. The maximum value is determined by the source impedance that the ADC will tolerate, typically 10K. If you go larger than this, you will erratic results. You tune the sensitivity by adjusting this value. Smaller values give less sensitivity.

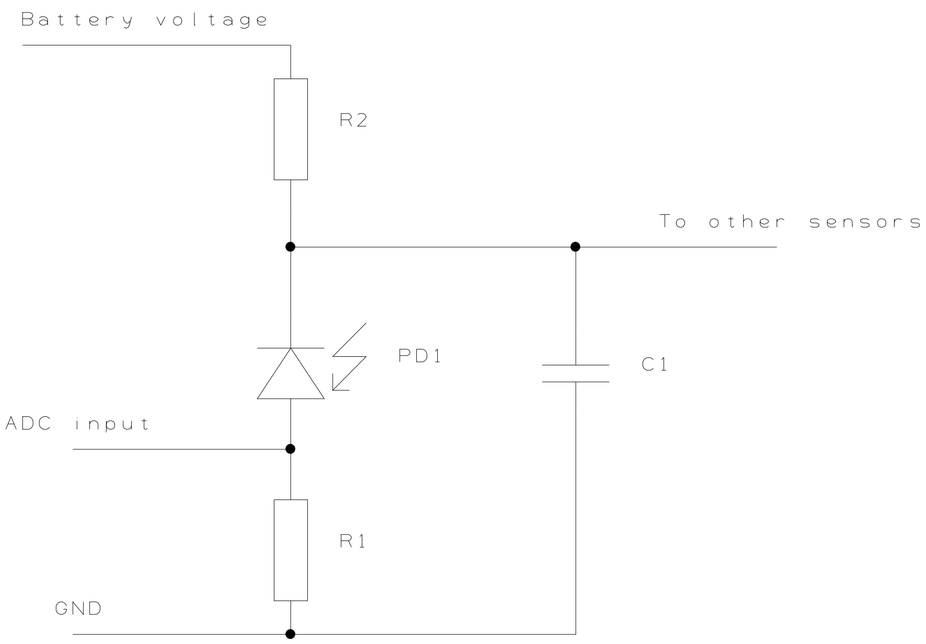
This has worked O.K. for me in the past but when I came to think about it for this paper I realised that it is flawed.

Photodiodes and phototransistors are specified at a particular value of reverse bias, typically 5V. When the above circuit is used, the photodiode is at ZERO reverse bias when full light level occurs. This is a non-linear region and gives maximum capacitance across the junction as well. It is to be avoided!

The solution is to supply the sensor circuit from a higher voltage or to use an ADC where the maximum range input is less than the supply voltage to the microcontroller. Some microcontrollers allow this but it does reduce the noise performance of the ADC.

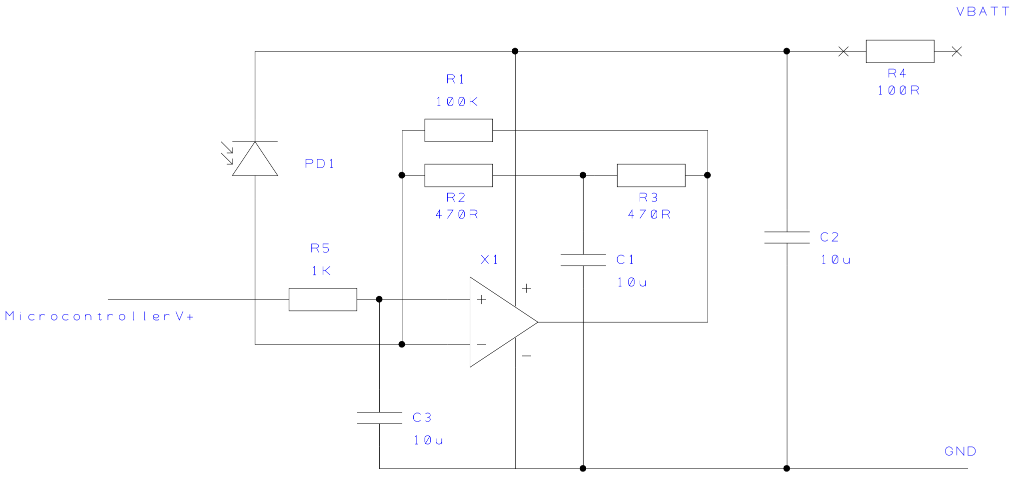
The higher voltage should be less than the rating of the device. They are typically rated at 20V so this should not be a problem.

The higher voltage doesn’t necessarily need to be regulated. The system that cancels ambient light will also cancel low-frequency variations of supply voltage.




The decoupling circuit can be shared between two or three sensors. R2 should be about one tenth of the value of R1 and C1 should be a ceramic type of 10uF or more. You may find that a small regulator supplying 6 or 7 volts is a cheaper/smaller alternative as it can feed 6 or more sensors. The current rating will need to be appropriate and for large phototransistors this can be several mA per sensor.

**More complex sensor**




Later I will be covering systems with very few sensors. Once there are two or less sensor channels, it can make sense to use better sensors with improved ambient light rejection.

Interference comes in various types. Sunlight changes at very low frequencies due only to clouds going over and the mouse movement. We can consider this as DC.

Artificial light often works at a higher frequencies. Fluorescent lights, incandescent lights etc. flicher at 100Hz in Europe and 120 Hz in America. This is more of pain.

LED lights can flicker at any frequency but it will be at least 100 Hz. This can be a pain. If the LED flicker is approaching the sense frequency of the mouse you can have a real problem. The only simple solution is some kind of blind. Fortunately, I haven't seen much of a problem with this. I suspect that lighting LEDs use a phosphor which filters the higher frequencies.

This circuit uses an op-amp to reduce the gain at ambient light frequencies and increase the sensitivity for the desired signal. The signal we are interested in is the first 10 microseconds or so of the LED pulse. We want to reject signals at frequencies lower than the signals corresponding to this time. The circuit has frequency-dependant feedback. At low frequencies, C1 behaves as an open circuit and the feedback resistor is effectively R2+R3; 940R. This behaves as the load resistor for the photodiode. At high frequencies, the effect of feedback through R2+R3 is bypassed to ground via C1 and the feedback resistor is effectively R1; 100K. This is 100 times bigger and gives, in theory, 100 times the gain.

So, theoretically, ambient interference at DC, sunlight, and mains frequency (100Hz in Europe, 120Hz in America) is reduced by a factor of about 100 relative to the deisred signal. The reality is not quite that good due to slew rate and gain bandwidth product limitations of the op amp. I tested a TLC272 and found that the response at 10KHz was 20 times that at 200Hz.

Because of the rise time of the op amp, you need a couple of microseconds delay between turning the LED on and reading the signal.

The op amp provides a low impedance source for the ADC so those issues disappear.

Note that the signal is inverted relative to the other systems so zero signal corresponds to the biggest ADC number and the ADC number reduces as the signal gets bigger. This means that smaller signals correspond to bigger reflections.

If you want two channels, use a dual op amp. You can share R4, R5, C3 and C2 for both channels. If you can get away with two channels of ADC, using a dual op amp would seem to be acceptable in terms of size, weight and cost.

**System stuff**

I have worked through the pure electronics, now I am going to discuss other system issues that should be considered.

**ADC resolution and speed**

When you select a microcontroller for micromouse, it is worth looking at the ADC specifications. Speed is straightforward, faster is better.

You need to be able to take a reading in less than 10 microseconds if you are using the simple driver. Even with the constant current version, you need to keep the conversions fast or the storage capacitor will have discharged before you take the reading. (Strictly speaking, it is the acquisition time that needs to be fast, but don't risk it.) If it takes you 300 microseconds to access, for example, a separate, serial interfaced, ADC, you will find life tricky.

The speed should include the influence of your chosen driver software. If you use a driver that initialises all of the ADC every time you use it, you may be very disappointed with the speed, even if the microcontroller is fast.

Common microcontrollers have resolutions of 10 or 12 bits. More is better provided it doesn't cost you in speed. Consider a setup where the ambient light is the same intensity as your light source at a range of 5mm. The setup would be adjusted for maximum ADC range or a bit less at this point. So, a 12 bit ADC would give you a count of 4095 here.

Half of this could be ambient so the signal would be a count of 2048. The ambient would be an offset of the same order and this would be subtracted from the total.

We have an inverse square law if all the light from the light source is reflected from the wall. In consequence, at 10mm range, the signal value is 1023, at 20mm, 511, at 40mm 255, at 80mm, 127, at 160mm, 63. A 10 bit ADC gives one quarter of these numbers, so you are talking about a signal value of about 15 at 160mm. This is pretty marginal, so 12 bit ADCs are recommended.

You can improve the resolution of the ADC. Simple digital signal processing theory says that doubling the number of samples, adds one bit of resolution. So, take two samples close together and you have made your 10 bit ADC into an 11 bit ADC. Four samples make it a 12 bit ADC.

All well and good, but it all takes time. I don't think you will be able to get more than four samples in a reasonable time. (10 microseconds or so). Don't forget that the 12 bit ADC can do the same, so four samples makes it a 14 bit ADC and all the sensor numbers are that much bigger and more useful.

My favoured microcontroller is an STM32F072 which has a 12 bit, 1 microsecond, ADC. Pretty useful!

Many drivers for ADCs (mine included), take a channel number and return a value. Setting the channel number is usually only a register write but it takes time. When you want to take several conversions on the same channel, you don't need to set the channel number each time. You could review the driver software to see if you make it faster by altering this.

For biggest signals with the simple LED driver and better power consumption for any driver, consider the following pseudo code:

```
SET CHANNEL NUMBER
TURN LED ON
READ ADC
TURN LED OFF
```

For an extra bit of resolution consider:

```
SET CHANNEL NUMBER
TURN LED ON
READ ADC
READ ADC
TURN LED OFF
ADD READINGS TOGETHER
```

This assumes that ADC setup is done elsewhere, probably at reset.

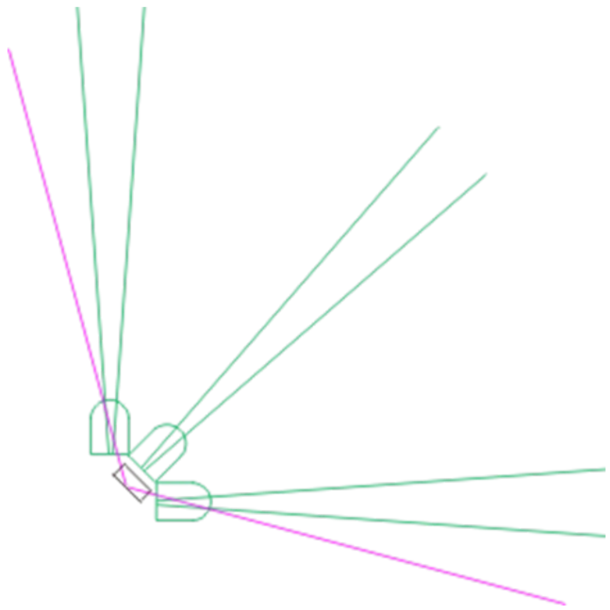
You may get screwed if you have an optimising compiler that thinks that both ADC reads are doing the same thing and decides it's quicker only to do it once!

**Multiplexing**

Some microcontrollers have a limited number of ADC channels. Some pre-assembled microcontroller boards have a limitation on the number of I/O pins available.

It is possible to get more sensor channels than the number of ADC channels.

If the sensors are wide angle, you can have more than one light source visible to the photodiode/transistor.



In the diagram, there are three narrow (4 degree) light sources pointing in roughly the directions that they might appear on one side of a mouse. The lines show the edges of the light beams.

There is also a single, wide angle (60 degree) light sensor. Again, the lines show the field of view.

A light spot from any of the light sources is within the field of view of the sensor.

The sequence:

LED1 ON, READ ADC, LED1 OFF

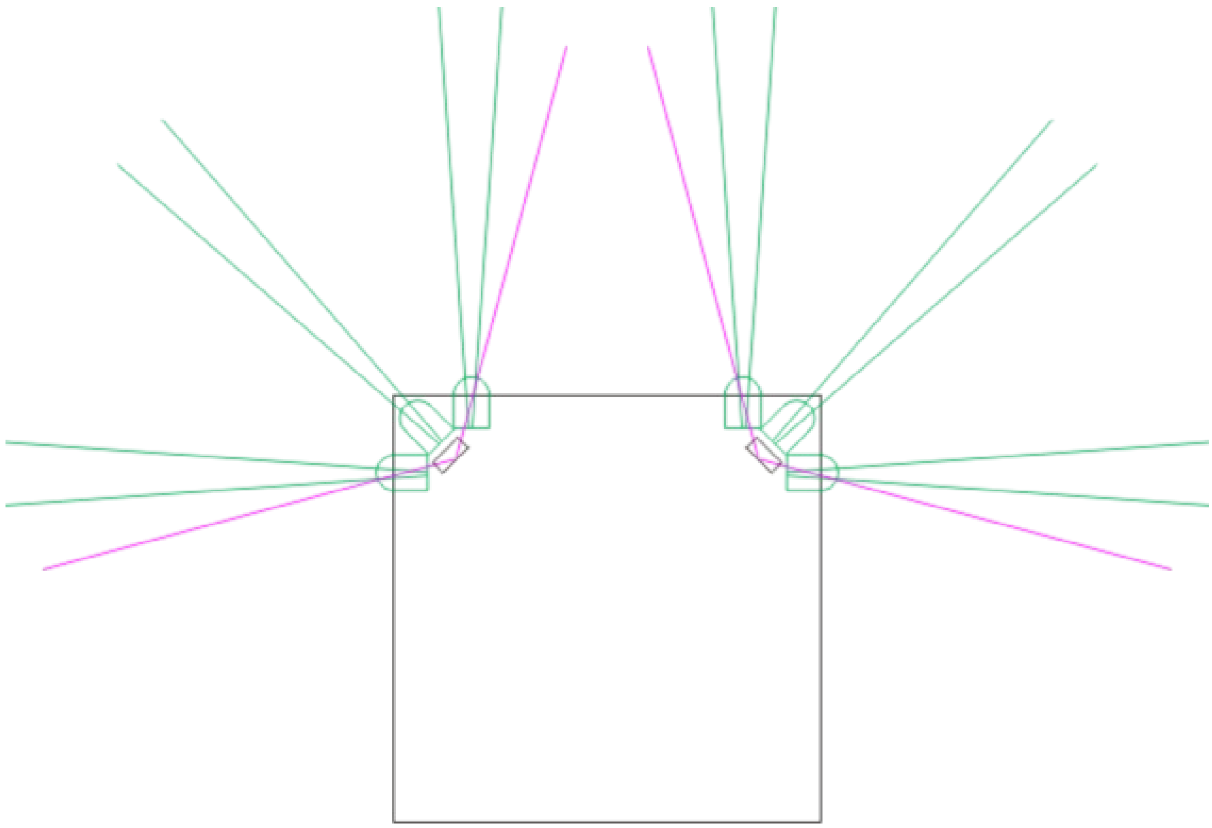
LED2 ON, READ ADC, LED2 OFF

LED3 ON, READ ADC, LED3 OFF

READ ADC \ ambient light level

will deliver the data to calculate the measurements for each of the three light sources.

You might need to read the ambient light between each LED “on” reading if the ADC is a bit slow. Provided that all the readings can be taken in less than about 20 microseconds this shouldn’t be needed.



You can go further. You could have two 3-channel sensors as above with one facing each side of the mouse. You can connect then the two photo sensors in parallel. Now you have to light the six LEDs in sequence and read the combined ADC input for each one.

The disadvantage of these systems is that the combined field of view of the sensors is larger than needed and more ambient light will be collected. Judicious optical screening of the sensors can reduce this.

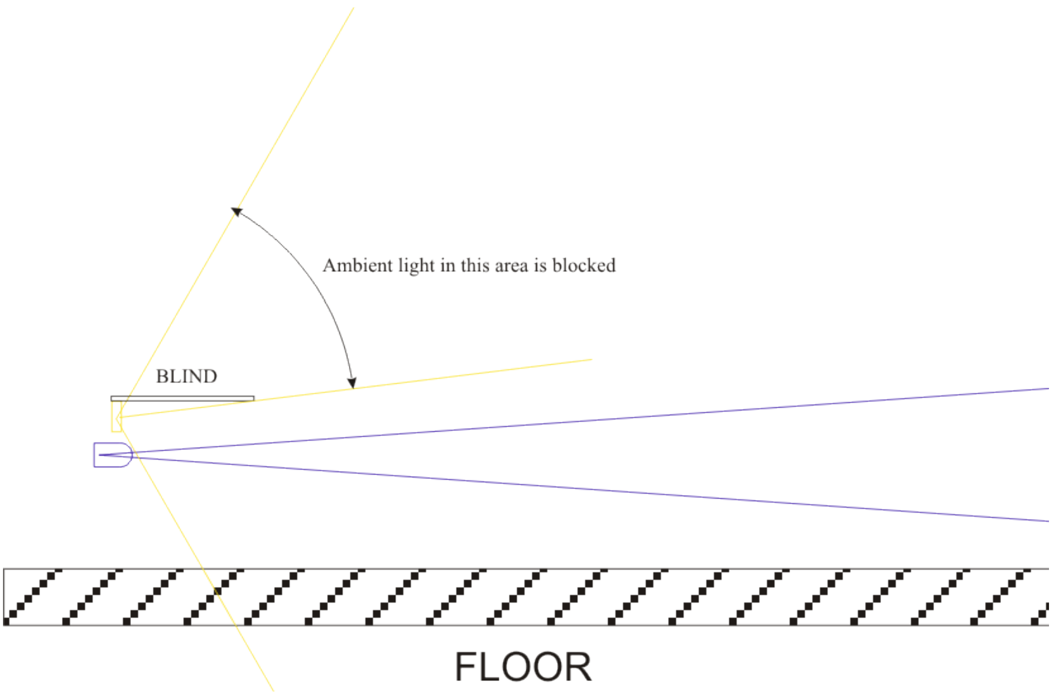
This needed a total of 7 I/O pins. You can use less.

Two LEDs can be turned on/off with the same I/O pin provided that each is outside the field of view of one of the sensors. They are both on at the same time but each sensor only sees one of them. This will give 6 sensor channels from 5 I/O pins.

**Managing the optics**

Wide angle sensors are good for multiplexing and making sure that all the reflected light from the wall enters the sensor, but they will collect more ambient light. Consider the use of blinds to cut down on the amount of ambient light that they can see. You can make a significant difference to the ambient light level your mouse will tolerate by doing this.

Note that it is also feasible to use mirrors (aluminium plated acrylic is light and easy) and lenses to modify the field of view to almost any desired pattern.




You will need to be careful of the material used for blinds. FR4, as used for PCBs, is not opaque. If you want to use part of your PCB as a blind, make sure that that part has continuous copper over it.

Some filament for 3D printing is opaque.

I know of no 3D printer resin that is sufficiently opaque to use as a blind.

You can also use paint to block off light. I have also had good results with Tippex! The drying time is really good.

**Linearity**

In theory, provided all the light from the source is reflected from the wall in such a way that all the reflection is in the field of view of the sensor, the ADC reading is proportional to the inverse of the square of the range.

Lots of things mess this up. One is direct light leakage from the light source to the sensor. This is comparatively straightforward to blank off with blinds. Light reflection from the floor can be an issue. Provided that you use a narrow light source, this should be minimised. If you let the wall come too close you may have problems with the sensor not seeing all the light.

I have had good success with suitable heat-shrink on the LED. It's quick and easy.

So, there may be linearity challenges at long and short range.

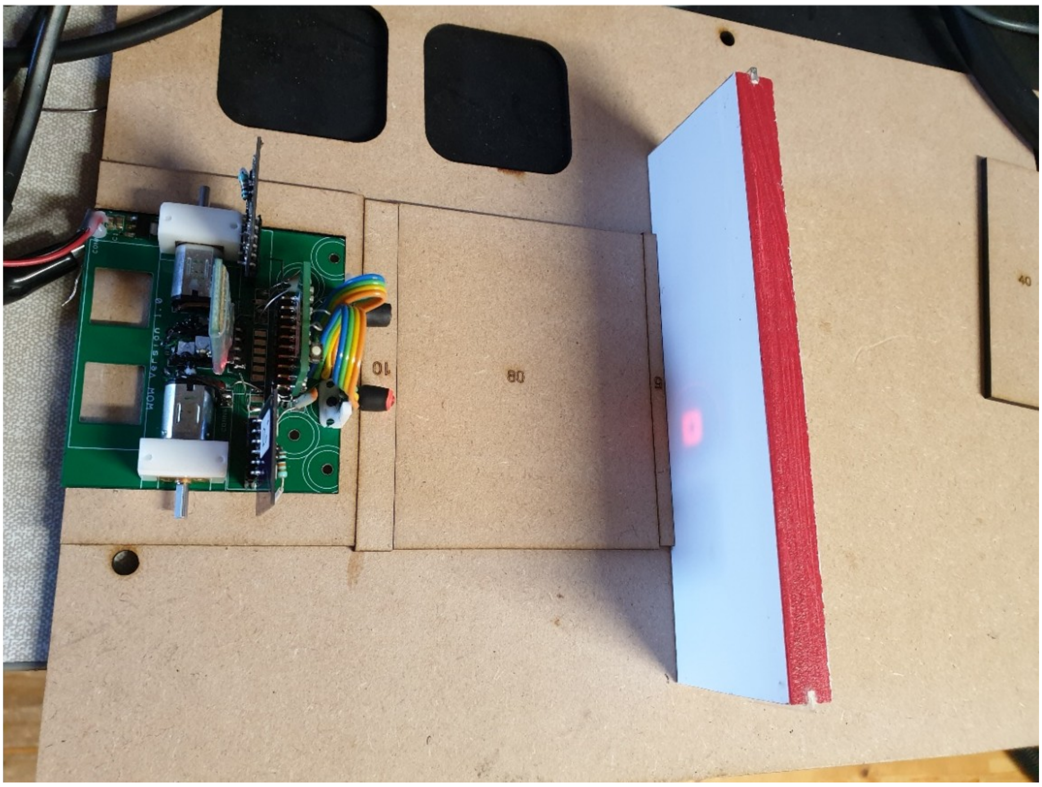
However, it is worth reviewing what we need the sensor to do. At long range, we are only interested in whether the wall is there or not. Linearity is not important, sensitivity and consistency are.

At short range, we only need to know that we are too close. Again, linearity is not important.

At intermediate ranges, we need good distance data to feed the mouse position PID controller. If the system is not linear, the gain of the controller system will vary depending on the distance from the wall. If the distance doesn't change much because your mouse is really well controlled, you may get away with just using the raw sensor data. With a good system, you may be able to keep the mouse position constant within 10mm or so and the sensors will be pretty linear over this range. Unfortunately, you may find this works fine except sometimes when the mouse finds itself 30mm out of position and goes unstable. If you have loads of processing power, doing the square root and inversion processes are probably worthwhile. You will only know how good the sensors are by testing and plotting the results.

**Testing**

I test a lot. I made a jig that holds the sensor in position and has a stop to put spacers against. The spacers are laser cut in widths of 5mm, 10mm, 20mm, 40mm, and 80mm. This allows me to position a wall at 5mm increments up to 155mm from the sensor. It holds the wall normal to the light which may give different results to the situation in a real maze as only some of your sensors will usually be at right angles to the wall. It does give useful comparisons though.



It's a good idea to characterise as many different walls as possible so that you know what you may have to cope with at a competition.

**Range tweaking**

You might find that you need to adjust the range of the sensor for your mouse. In every case you should design the light source for maximum light output to ensure best signal to ambient ratio. You should tune the range by reducing the sensor sensitivity as needed by adjusting the load resistor to suit.

**Which way should the sensors point?**

In some ways, I don't know. In others I have recommendations.

One thing to try and avoid, is to have the sensors pointing at right angles to the wall. If you think of the wall as a mirror, you don't want the mirror reflection to fall back on the sensor as you will end up with a much better angle sensor than range sensor!

One generalised way to avoid this is for the light sources to point slightly up or down. That way the reflected image will miss the light detector for every orientation of the mouse relative to the wall. Mounting the sensors high or low on the mouse helps to avoid the problem.

There may be issues if you want to do this for a long-range sensor. You can't tilt the light source much without it missing the wall at long ranges. If you make sure that the sensor and light source are as close together as possible you minimise this. The sensor doesn't have to be above or below the light source, it can be at the same level so this helps.

Sometimes, you want the maximum distance at which you can see the wall to be as well defined as possible. Sometimes, the reflection of walls in adjacent cells can cause confusion. You might want good sensing out to 180mm and then as sharp a cut-off as possible after that. It can help if you arrange the light source so that it starts to spill over the wall or onto the floor at this range. Once the light spot is entirely over the wall or entirely onto the floor, the reflection should be negligible.

You have the choice of tilting the light source up or down. I prefer tilting it down as it avoids issues with white objects near the maze. Someone could be standing next to the maze in a white T shirt and give a strong reflection. The floor is at least fairly consistent.

Mounting the sensors high or low allows the biggest tilt. Mounting high raises the centre of gravity but allows the sensors to be angled downwards. Mounting low improves the centre of gravity but mandates that you look towards the top of the wall with the consequent risk.

Where the sensors are mounted and where they point relative to the mouse are interesting subjects. I have no definitive answers. Some people like the forward-facing sensors to be mounted on the corners of the mouse pointing slightly outwards so that the mouse can do "obstacle avoidance" when doing a diagonal run.

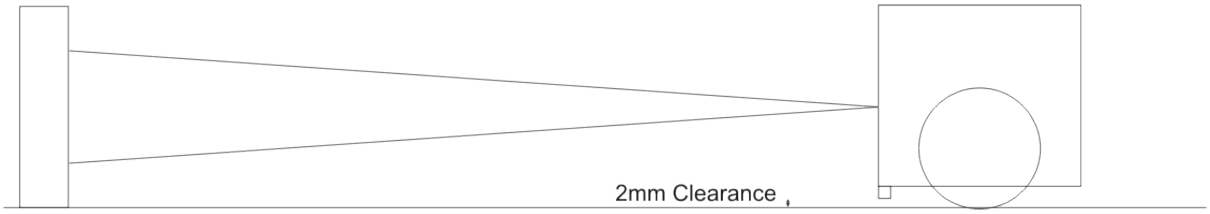
Some people like a set of sensors pointing at right angles to the mouse for accurate detection of the position of gaps in the walls. One of my mice had sensors mounted such that when they lost the wall, it was exactly the right point to start a smooth turn.

The angle of the sensors used for positioning the mouse and detecting walls is a moot point. The smaller the angle to the wall, the more accurate the control loop can be but you see a lot of other walls at certain points in the maze which limits the percentage of each cell that you can use to correct the mouse position. If you use too sharp an angle, you run the risk of the control loop going "over centre" when the mouse is incorrectly pointed. If the sensor points directly at the wall, turning either way makes the sensor read a bigger number so this "over centre" effect can be catastrophic!

I have tended to use too small an angle, so don't copy my mice here!

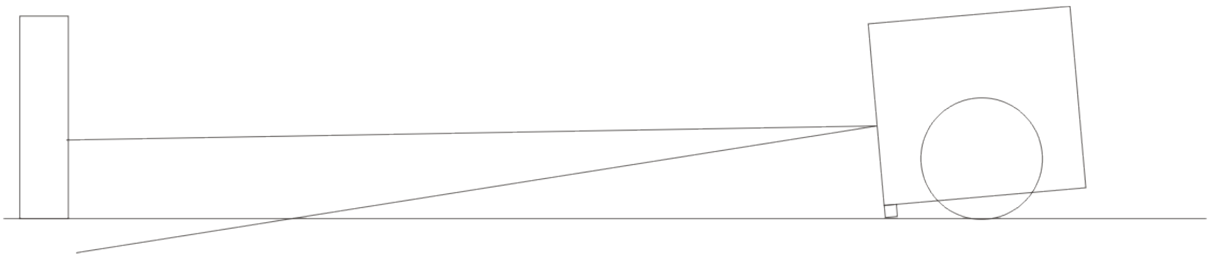
**Mouse tilt**

The mouse will typically be a “wheelchair” arrangement with two driven wheels and a slider. There will also be something (intentional or not!) that limits the rotation of the mouse under fierce braking or acceleration as the slider is lifted off the ground. There has to be clearance between this limiter and the floor so that the mouse doesn’t get “beached” ate floor joints or changes in level. The clearance could be of the order of 2mm.




The diagram shows the sensor aligned to the centre of the wall at full clearance.

When braking hard, the mouse will tilt forward.



As you can see, at longer range, the light spot dramatically changes its position on the wall. This will change the reflective signal greatly.

You should consider whether the mouse should be front-heavy or back-heavy and design the sensor direction accordingly. I tend to favour front-heavy so that the light spot position is correct under braking as this is the condition when you most want to see the front wall accurately as you may be positioning for a turn using the data. During fierce acceleration, you probably already know that there is no wall in the way anyway.

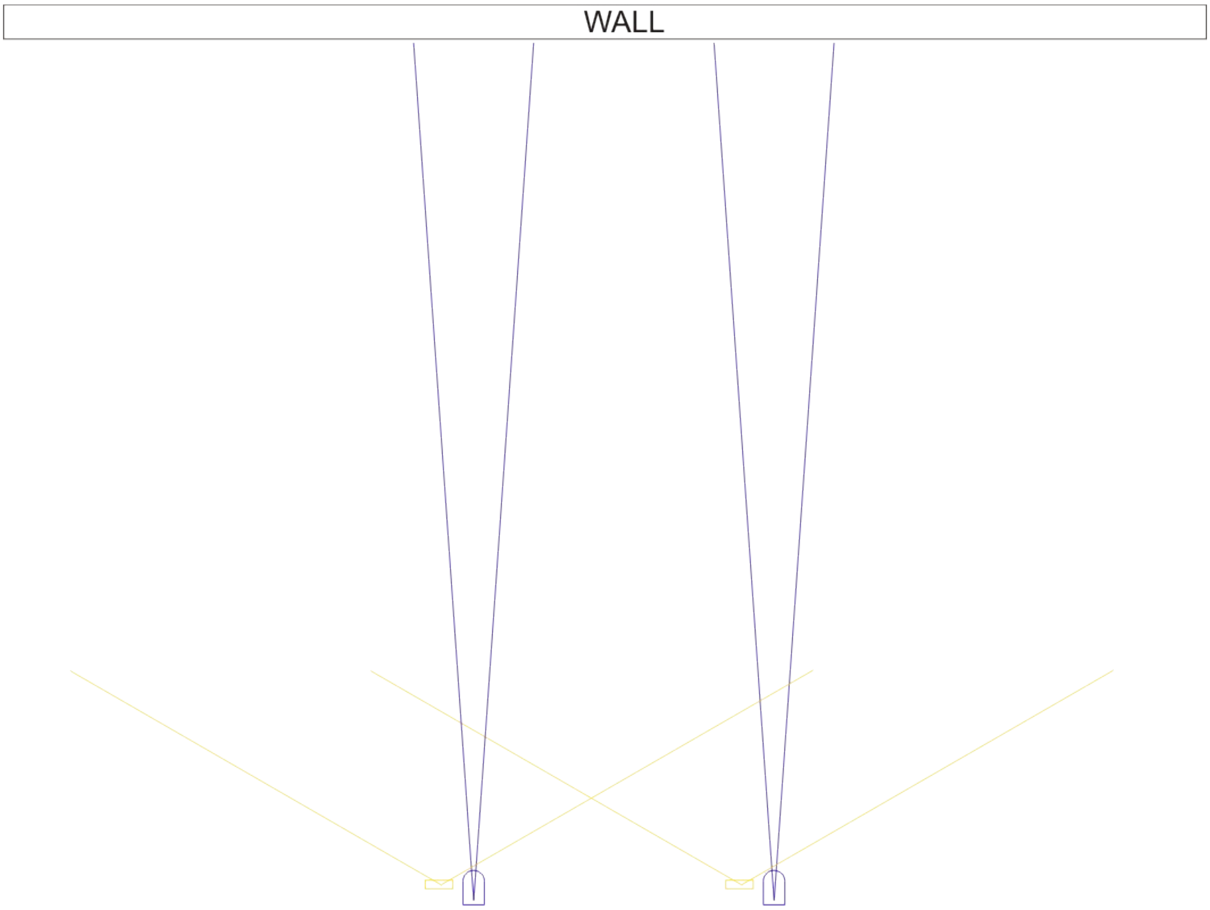
**Range extension**

Sometimes, you might want to detect a wall at the greatest possible range. There are ways to do this, but there are inevitable trade-offs. One simple method is, as previously described, take multiple readings and add them together. The pulsed light source might limit how much of this is possible.

You could increase the light output by using multiple LEDs for each sensor. If you use the classical pulsed LED, you could have two or more LEDs in parallel with separate current limiting resistors and storage capacitors but operated by a single switch device. Doubling the number of LEDs will have a similar effect to doubling the number of readings. The power consumption will increase accordingly.

With my new constant current driver you can use LEDs in series to give the same effect. You will need to ensure that there is enough “headroom” in terms of battery supply voltage. I find that I can operate two LEDs in series from a 9 V supply with no issues. This gives a 40% or so range increase at the cost of just one LED per sensor.

An alternative for detecting front wall at long distance would be to use a variant on the multiplexing scheme described above. If both the front-facing sensor fields of view can see the reflections of both the light sources, you can turn both sources on while you read the values of each sensor. Each sensor sees both LEDS and gets about twice the light. You can add the readings from both sensors together to double the signal again. Provided that none of the light has spilled off the walls, the signal has increased by about a factor of four to give a range extension of about double for free!




Don't expect it to be particularly linear but if it enables you to use a wall detection as a braking point for a high speed run, it could be quite valuable. Extending the range of sideways-looking sensors will need careful thought. You may be able to reduce search times by looking for walls in nearby cells as you travel past but you might get confused by detecting them. You should always be within 100mm or so of the walls in the current cell unless the mouse has done something crazy!

**Final Comments**

Sensor mountings must be rigid and strong. The sensors must not change position when the mouse hits the wall (again) otherwise you will get very confused when tuning the mouse. This is tricky as you want to be able to adjust the sensors to fine tolerances during testing. I usually end up using hot melt glue to fix the position when (if) I am happy with the results. You can always reflow the glue using the glue gun.

You may have spent months developing a perfect sensing system and now it is time for the competition. You have a set of wonderful precision optical instruments. Keep them clean! Fingerprints etc. will have similar effects on the sensors as they do on your spectacles.

For simple wall- and line-followers you can often find that you can solve any particular issue by changing the sensors, electronics, mechanics or software. If you want to be the best, find the right fix. Don't get obsessed with one particular type of solution.

If you have followed all this, you should be able to make some reasonable sensors. That just leaves everything else!