# Using BBC micro:bits with light gates

A close up of a machine

Description generated with high confidence

**Overview**

This article accompanies and gives instructions for the six micro:bit light gate programs on the SSERC website. All six programs were written in micropython using the mu editor.

Programs 1 and 2 are ‘master’ programs. They can be used on their own. They differ only in what information is displayed – program 1 displays the gate time i.e. how long the light beam was broken; program 2 uses the gate time and assumes a single mask of 10 cm width and thus displays the calculated speed.

Programs 3 and 4 are ‘slave’ programs. They need to be triggered by a ‘master’ program and as such cannot be used on their own. Programs 3 and 4 differ only in what information is displayed – program 3 displays the gap time i.e. the time between the midpoint of the master pulse to the midpoint of the slave pulse. Program 4 displays the gap time and the speed at the slave light gate (again based on a single 10 cm wide mask).

Programs 5 and 6 use an asymmetrical double mask (Fig 11.) to measure acceleration through a single light gate. Program 5 displays *u, v, t* and *a*. Program 6 displays only the acceleration.

Download the .zip file onto your computer and extract the hex files. Connect your micro:bit to your computer. It should appear as a new drive. Load the desired program onto your micro:bit by dragging the hex file onto the micro:bit.

**Programs 1 and 2**

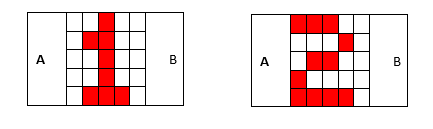
To avoid possible confusion when using several micro:bits in a single experimental setup, each program displays its program number at startup (see Figs 1 &2 below).

Fig 1. Fig 2.

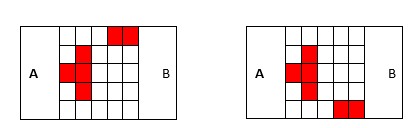
Pushing Button A on each micro:bit takes you to a setup screen (see Figs 3 & 4 below)

Fig 3. Fig 4.

This allows you to set up your light beam, mask and light sensor for each micro:bit by looking at the two LEDs on the right hand side of the display – an uninterrupted light beam should lead to the two LEDs showing HIGH (Fig 3). An interrupted light beam should show LOW (Fig 4). We found that a tubular shroud of length 2 cm, made of black paper, helped isolate the light sensors from stray light.

When you are happy with the setup and ready to start measuring, push Button A (as suggested by the small arrow showing in the left hand side of the display).

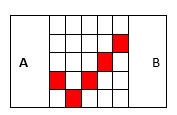
You will now see the ‘ready’ indicator ( a ✓, see Fig 5).

Fig 5.

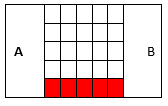
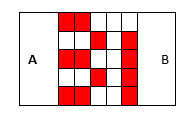
When the light beam is Interrupted the display will change to LOW (Fig 6).

Fig 6.



When the light beam once again reaches the sensor the measurement(s) displayed by the program are scrolled across the screen (Fig 7 below).

Fig 7.

**Master**

When the light beam is interrupted and also when it again reaches the light sensor Programs 1 and 2 broadcast a radio signal to any listening ‘slave’ micro:bits, allowing a gap time to be measured by micro:bits running programs 3 or 4.

**Reset**

When all micro:bits are displaying measurements, pushing Button B when running ‘master’ programs 1 or 2 will return the micro:bit AND any micro:bits running a ‘slave’ program directly back to the ‘ready’ state, awaiting further measurements. Pushing the ‘reset’ button on the back of the micro:bit will start the whole ID and setup process from the start.

**Programs 3 and 4**

Again to avoid confusion when using several micro:bits, each program displays its program number at startup. (See Figs 8 and 9 below)

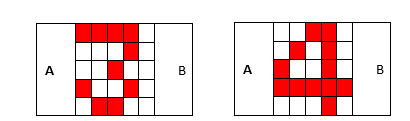


Fig 8. Fig 9.

Again pushing Button A takes you to a set up screen as shown in Figs 3 and 4 allowing you to align light beam, mask and light sensor. Push Button A when ready to measure and you will be taken to the ‘ready’ screen as in Fig 5.

Micro:bits running programs 3 and 4 will remain in this state until they receive a radio message to say that a nearby master has transitioned from HIGH to LOW and a second radio message to say that the nearby master has transitioned from LOW to HIGH. When both radio messages have been successfully received the display will change to show the current measured light level (HIGH or LOW) and programs 3 and 4 now monitor the light sensor to detect level changes, timestamping these events.

The measurements from the selected program will then be scrolled across the display.

The program can be returned to the ‘ready’ state by pushing button B on the master micro:bit.

Pushing the ‘reset’ button on the back of the micro:bit will start the whole ID and setup process from the start.

**Hardware**

The light level voltage is measured on pin1 of the micro:bit.

A light dependant resistor and a 10kΩ resistor will be needed for each micro:bit (See Fig 8 and 9).

Alternatively a readymade unit (excluding micro:bit!) can be purchased for approximately £3 (July 2017) (See Fig 10).

micro:bit 3V

micro:bit pin1

micro:bit GND

10 kΩ

ORP12

Fig 8.

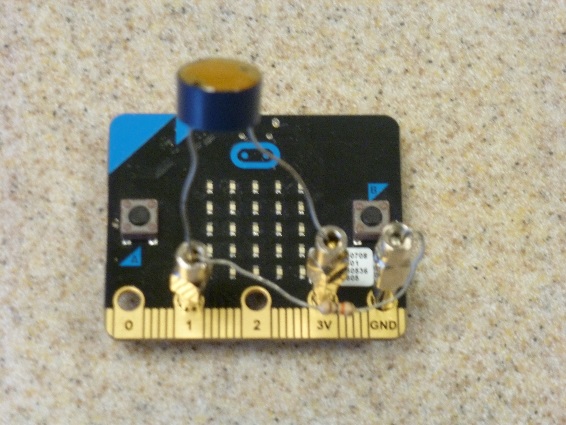


Fig 9.

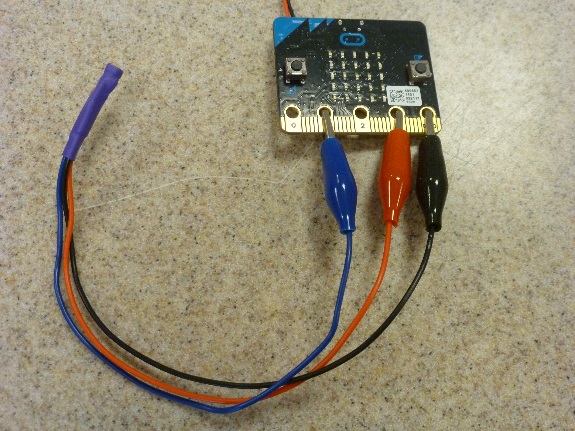


Fig 10.

**Limitations**

This is not the equipment to use in an Advanced Higher investigation to find g to three significant figures. It would appear that the smallest unit of time that can be measured with the micro:bit using standard micropython and the mu editor is 6 ms. To avoid large uncertainties, a 10 cm mask is used with programs 2, 3 and 4, rather than a shorter 4 cm mask that might be used with a commercial timer such as a DJB TSA. Similarly, programmes 5 and 6 utilise the unusual asymmetric mask shown in Figure 11. For acceleration investigations, it is best to start with the mask as close to the light gate as possible so that the initial time measurement is relatively large. We were able to investigate the relationship between slope angle and acceleration for slopes up to around 10 degrees.

Should you wish to develop this software in C, which we think could give more accurate timing perhaps to the level of being able to measure the speed of sound in air, we will happily supply the .py versions of the program. Equally if you are in the (admittedly unlikely) position where you can only find a 9.5 cm mask and they wish to change the default card length in the .py script using the mu editor or even just want to change the light level threshold value you are again welcome to the .py script – just get in touch.

**Asymmetrical Mask**

The Asymmetrical Mask shown in Fig 11. below is made from a single sheet of A4 card. The narrow mask is 5 cm wide the wider mask 20 cm. The length of the A4 card is 296 mm. Choose the height to suit your situation. Drop the narrow mask first through light gate.

50 mm

200 mm

Fig 11.

296 mm

Summary of programs

In all the set-ups below, the micro:bit is connected to one of the light sensors shown in Figures 9 and 10. We used inexpensive LED torches as our light sources.

|  |  |  |
| --- | --- | --- |
| Program | Function | Possible uses |
| 1  (master) | Measures the time the light beam is broken. Displays this time. Also sends a timing signal to any micro:bit nearby that is running programs 3 or 4 | The displayed time can be used by pupils to find instantaneous speed if the length of the mask is known. |
| 2  (master) | Measures the time the light beam is broken. Assuming a 10 cm mask is used, displays the instantaneous speed at the light gate. Also sends a timing signal to any micro:bit nearby that is running programs 3 or 4. | Any investigation where instantaneous speed is required. See notes for programs 3 and 4 below too. |
| 3  (slave) | When used with another micro:bit that is running a master program, displays the time between a mask cutting the light gate at the master and cutting it at the slave. | Average speed between the master and slave can be calculated using the time on the slave and the distance between the two sets of light gates.  More than one slave can be used, so average speed from the first gate can be found at several points on a journey. |
| 4  (slave) | When used with another micro:bit that is running a master program, displays the time between a mask cutting the light gate at the master and cutting it at the slave. Also displays the instantaneous speed through the slave light gate. | Calculation of acceleration when used with a micro:bit running program 2. The master will give the instantaneous speed at its own light gate, whilst the slave will display speed at the slave and the time from the master to the slave (times are calculated from the midpoints of those used to calculate the instantaneous speeds). |
| 5 | When used with the mask shown in Figure 11, displays u, v, a and t when the mask cuts a light beam. | Calculation of acceleration |
| 6 | When used with the mask shown in Figure 11, displays acceleration when the mask cuts a light beam. | Investigation of the variation of acceleration with slope angle, etc. |