Name: _____

Assessment

Beam Break

1) When calibrating the Beam Break Sensors for accurate speed measurement, what critical factors should you consider?
    a) The color of the car and the ambient light conditions
    b) The distance between the sensors and the height of the car
    c) The type of material used for the track and the weight of the car
    d) The power supply voltage and the length of the jumper wires

2) What is the purpose of using infrared light in the Beam Break Module?
    a) To illuminate the track
    b) To power the sensors
    c) To signal the car to start
    d) To detect the car passing through

3) What do the Beam Break Sensors measure to calculate the car's speed?
    a) Temperature change
    b) Light intensity
    c) Distance traveled
    d) Time difference

4) Which function in the program calculates the time between the first and second beam breaks?
    a) Beam break time
    b) Ready set go lights
    c) Traffic light control
    d) Start car speed

5) How is the car's speed finally displayed in the program?
    a) On a traffic light
    b) On a digital clock
    c) In the console
    d) On an LED screen

6) Explain how you could use Beam Break Sensors in a real-world scenario outside of measuring car speed.

© 2024 Play Piper, Inc.

Name: _____

7) Imagine you want to improve the accuracy of measuring Piperbot's speed. What could you do to achieve a more accurate measurement?

8) Look at the provided Python code. How could you modify this code to add a yellow light that blinks twice before the green light turns on? Write the new code with your changes.

© 2024 Play Piper, Inc.

Name: _____

Answer Key

Beam Break

1) B - The distance between the sensors and the height of the car

2) D - To detect the car passing through

3) D - Time difference

4) A - Beam break time

5) C - In the console

6) *Example:* Beam Break Sensors can be used in automated door systems to detect when someone enters or exits a building. By placing sensors on either side of a doorway, the system can detect the interruption of the beam, signaling the door to open or close.

7) *Example:* Because real-world measurements always have some variance, you could take multiple measurements and average the result. Additionally, you can turn off the Digital View by clicking on the Digital View tab at the bottom of the screen and toggling the switch before starting the code. This helps the Pico run faster (because it doesn't have to take time to report its pin states to the computer), which will result in better resolution (higher accuracy) for the speed measurements.

8) Example:

```
## ---- Imports ---- ##
import time
import board
from piper_blockly import *
from digitalio import Pull

## ---- Definitions ---- ##
first_beam_time = None
second_beam_time = None
car_speed = None
GP13 = piperPin(board.GP13, "GP13")

try:
  set_digital_view(True)
except:
  pass

GP6 = piperPin(board.GP6, "GP6")
```

© 2024 Play Piper, Inc.

Name: _____

```
_clock_start = time.monotonic() + 0.09 # adjust for startup time
def chip_clock():
  global _clock_start
  return time.monotonic() - _clock_start

GP11 = piperPin(board.GP11, "GP11")
# Describe this function...
def beam_break_time():
  global first_beam_time, second_beam_time, car_speed
  while not ((GP6.checkPin(Pull.UP))):
    pass
  first_beam_time = chip_clock()
  while not ((GP11.checkPin(Pull.UP))):
    pass
  second_beam_time = chip_clock()
  return second_beam_time - first_beam_time

GP15 = piperPin(board.GP15, "GP15")
GP14 = piperPin(board.GP14, "GP14")
# Describe this function...
def ready_set_go_lights():
  global first_beam_time, second_beam_time, car_speed
  GP15.setPin(1)
  time.sleep(2)
  GP15.setPin(0)
  GP14.setPin(1)
  time.sleep(0.5)
  GP14.setPin(0)
  time.sleep(0.5)
  GP14.setPin(1)
  time.sleep(0.5)
  GP14.setPin(0)
  time.sleep(0.5)
  GP13.setPin(1)


## ---- Code ---- ##
while True:
  ready_set_go_lights()
  car_speed = 0.179 / beam_break_time()
  print(str(car_speed) + ' miles per hour')
  GP13.setPin(0)

  time.sleep(0.1)
```

Explanation:
In this code, the yellow light connected to GP14 is modified to blink twice before the green light turns on. The yellow light is turned on for 0.5 seconds, turned off for 0.5 seconds, and repeated to achieve the blinking effect.

© 2024 Play Piper, Inc.