**NZQA**

**Approved**

EXPIRED

Achievement standard: 91078 Version 3

Standard title: Implement basic interfacing procedures in a specified electronic environment

Level: 1

Credits: 3

Resource title: Stop thief!

Resource reference: Digital Technologies VP-1.48 v2

Vocational pathway: Primary Industries

|  |  |
| --- | --- |
| Date version published | February 2015 Version 2  To support internal assessment from 2015 |
| Quality assurance status | These materials have been quality assured by NZQA.  NZQA Approved number A-A-02-2015-91078-02-7346 |
| Authenticity of evidence | Assessors/educators must manage authenticity for any assessment from a public source, because learners may have access to the assessment schedule or exemplar material.  Using this assessment resource without modification may mean that learners’ work is not authentic. Assessors/ educators may need to change figures, measurements or data sources or set a different context or topic to be investigated or a different text to read or perform. |

Vocational Pathway Assessment Resource

Achievement standard: 91078

Standard title: Implement basic interfacing procedures in a specified electronic environment

Level: 1

Credits: 3

Resource title: Stop thief!

Resource reference: Digital Technologies VP-1.48 v2

Vocational pathway: Primary Industries

Learner instructions

# Introduction

This assessment activity requires you to implement basic interfacing procedures to make a working sensor-controlled detection device to protect farm vehicles from theft.

You are going to be assessed on how efficiently you implement basic interfacing procedures to make a working sensor-controlled detection device to protect farm vehicles.

The following instructions provide you with a way to structure your work to demonstrate what you have learnt to allow you to achieve success in this standard.

Assessor/educator note: It is expected that the assessor/educator will read the learner instructions and modify them if necessary to suit their learners.

# Task

You will make a working sensor-controlled detection device to protect farm vehicles from being stolen, using basic electronic components and a microcontroller.

Basic ‘interfacing procedures’ refer to the selection, testing and debugging of the hardware and software that allows different devices to work together to meet the given specifications for the sensor-controlled detection device.

You will present a portfolio which will include evidence of how you use datasheets or calculations to help choose appropriate components (types and values) for the interface. Include in your portfolio evidence of the writing of your well-structured, functional and clearly annotated interface software.

## Specifications

Specifications must be agreed with the assessor/educator prior to the implementation of the interfacing procedures.

Specifications for the working sensor-controlled detection device include:

* a sensor device that operates at all times of the day, whether dark or light
* a sensor system suitable for detecting the movement of a vehicle or a person
* a remote indicator, using a LED, which shows when a vehicle or a person is in the monitored location
* a short warning buzzer that buzzes for 10 seconds to warn the farmer that there is a vehicle or person in the monitored location
* a system that starts a video camera to record the vehicle or person
* a circuit that provides a switch so the farmer can manually operate the detection device
* a switch for manually operating the detection device.

## Steps for implementing and testing

Choose appropriate component types and values for the working sensor-controlled detection device and test and debug a functional model.

Use data sheets or calculations to assist in choosing appropriate component types and values for the working sensor-controlled detection device.

Write well-structured, clearly annotated, and readily understandable software.

Select appropriate electronics components for an interface that links the working sensor-controlled detection device to the microcontroller. Show how you have used data sheets or calculations to assist in selecting these components. You may use circuit diagrams to assist in demonstrating this.

Write a computer program that controls the detection device and allows it to work according to the specifications provided. The software interface needs to be well structured, readily understandable, and clearly annotated. This means your computer program should:

* be clearly set out and correctly indented
* include comments that explain exactly what the program is doing at each step
* use labels so that it is easy to read and understand the program.

Test and debug a working model of the sensor-controlled detection device. This means checking that the sensor-controlled detection device works according to the specifications and taking corrective action where it is not performing as expected.

## Performing and completing the task

As you perform the task, make notes and gather evidence for inclusion in your portfolio. Take care to record in a journal or log the steps you take to create your working sensor-controlled detection device.

## Presentation

Present your evidence as a portfolio, which may include:

* annotated diagrams and photographs
* short video clips
* journal entries that describe testing and debugging activities
* your responses to assessor/educator questions
* interview notes from any scheduled assessor/educator consultation
* code for your computer program.

# Resources

Use the circuit below, if required.

## Useful websites

<http://www.arduino.cc/en/>

[www.electronics-tutorials.com/basics/basic-electronics.htm](http://www.electronics-tutorials.com/basics/basic-electronics.htm)

<http://williamson-labs.com/>

[www.technologystudent.com](http://www.technologystudent.com)

[www.picaxe.com](http://www.picaxe.com)

Vocational Pathway Assessment Resource

Achievement standard: 91078

Standard title: Implement basic interfacing procedures in a specified electronic environment

Level: 1

Credits: 3

Resource title: Stop thief!

Resource reference: Digital Technologies VP-1.48 v2

Vocational pathway: Primary Industries

Assessor/Educator guidelines

# Introduction

The following guidelines are supplied to enable assessors/educators to carry out valid and consistent assessment using this internal assessment resource.

As with all assessment resources, education providers will need to follow their own quality control processes. Assessors/educators must manage authenticity for any assessment from a public source, because learners may have access to the assessment schedule or exemplar material. Using this assessment resource without modification may mean that learners' work is not authentic. The assessor/educator may need to change figures, measurements or data sources or set a different context or topic. Assessors/educators need to consider the local context in which learning is taking place and its relevance for learners.

Assessors/educators need to be very familiar with the outcome being assessed by the achievement standard. The achievement criteria and the explanatory notes contain information, definitions, and requirements that are crucial when interpreting the standard and assessing learners against it.

# Context/setting

This activity requires learners to efficiently implement basic interfacing procedures to make a working sensor-controlled detection device to prevent theft of farm vehicles, using basic electronic components and a microcontroller. The sensor-controlled detection device must perform to the specifications outlined in the task.

# Conditions

This is an individual assessment task.

# Resource requirements

Learners should be provided with:

* access to a computer
* a microcontroller system that includes a simple switch and a light sensor that can detect the position of the vehicle or person, and is able to control a relay
* access to a digital camera and/or video camera to photograph portfolio evidence.

The list of resources may be adapted to meet the needs of your learners. Some learners may have cameras on their mobile phones. These can be used to document the process.

Electronic catalogues and supplies are available at:

[www.Surplustronics.co.nz](http://www.Surplustronics.co.nz)

[www.Activecomponents.com](http://www.Activecomponents.com)

[www.Mailtronics.co.nz](http://www.Mailtronics.co.nz)

[www.electroflash.org.nz/](http://www.electroflash.org.nz/)

Learners will also require:

* internet access
* materials to build their detection system
* an electronic breadboard
* electronic circuit simulation software, for example:

[www.falstad.com/circuit/](http://www.falstad.com/circuit/)

[www.yenka.com](http://www.yenka.com)

software for schematic capture, for example: [www.cadsoft.de](http://www.cadsoft.de)

The following resources may also be useful:

* *An Introduction to Practical Electronics, Microcontrollers, and Software Design* by Bill Collis (available at [www.techideas.co.nz](http://www.techideas.co.nz))
* *PICAXE Microcontroller Projects for the Evil Genius* by Ron Hackett (available at [www.amazon.com](http://www.amazon.com)).

# Additional information

The specifications need to be agreed prior to the implementation of interfacing procedures. They may be provided or developed in negotiation with the learner. When developing specifications, it is essential that:

* the specifications relate to the monitoring and control of variables in both hardware and software
* the electronic environment includes a functional combination of hardware and embedded software
* the specifications define the functional qualities required.

Learners are required to collect portfolio evidence as they complete the task. Guidance may be provided around what the evidence should look like.

Other microprocessors, such as Arduino could be used.

## Other possible contexts for this vocational pathway:

* a detector that signals when the sawdust needs changing in an animal pen
* a detector that starts a fan when it gets too hot
* a detector than measures frost temperature and starts the sprayers
* a detector that opens a gate automatically
* an animal feeder that feeds when a certain light level is reached
* a detector that detects when the voltage in an electric fence is not optimum.

# Assessment schedule: Digital Technologies 91078 – Stop thief!

|  |  |  |
| --- | --- | --- |
| Evidence/Judgements for Achievement | Evidence/Judgements for Achievement with Merit | Evidence/Judgements for Achievement with Excellence |
| The learner implements basic interfacing procedures in the sensor-controlled detection device by:   * choosing appropriate component types and values for the interface   For example:  *I found that the LDR had a resistance of 68 kΩ when the light beam was blocked and 4 kΩ when the light beam was not blocked. So I chose a resistor in between these two values to use in the voltage divider. I used a 30 kΩ resistor.*     * writing basic functional interface software given simple programme structures   For example:  See Appendix 1 for an example illustrating different versions of a code fragment.   * testing and debugging a functional model of the interface for the sensor-controlled detection device   For example:  *I found that when the sensor was facing into the room, the detection device worked perfectly, but when I turned it around so it was facing the windows, it did not work anymore. I fixed this by making a little shield for the light sensor so that sunlight could not land on it. I also found that the light sensor worked OK one day but not the next. This was because the brightness of the light in the room changed. I fixed this by shining an LED onto the sensor and hiding the sensor from any other light. This made the sensor-controlled detection device work with any lighting conditions (see my annotated photographs of this).*  *The above expected learner responses are indicative only and relate to just part of what is required.* | The learner skilfully implements basic interfacing procedures in the sensor-controlled detection device by:   * choosing appropriate component types and values for the interface   For example:  *I found that the LDR had a resistance of 68 kΩ when the light beam was blocked and 4 kΩ when the light beam was not blocked. So I chose a resistor in between these two values to use in the voltage divider. I used a 30 kΩ resistor.*     * writing annotated, functional, and readily understandable interface software given simple programme structures   For example:  See Appendix 2 for an example illustrating different versions of a code fragment.   * testing and debugging a functional model of the interface for the sensor-controlled detection device   For example:  *I found that when the sensor was facing into the room, the detection device worked perfectly, but when I turned it around so it was facing the windows, it did not work anymore. I fixed this by making a little shield for the light sensor so that sunlight could not land on it. I also found that the light sensor worked OK one day but not the next. This was because the brightness of the light in the room changed. I fixed this by shining an LED onto the sensor and hiding the sensor from any other light. This made the sensor-controlled detection device work with any lighting conditions (see my annotated photographs of this).*  *The above expected learner responses are indicative only and relate to just part of what is required.* | The learner efficiently implements basic interfacing procedures in the sensor-controlled detection device by:   * choosing appropriate component types and values for the interface   For example:  *I found that the LDR had a resistance of 68kΩ when the light beam was blocked and 4kΩ when the light beam was not blocked. So I chose a resistor in between these two values to use in the voltage divider. I used a 30kΩ resistor.*     * writing well-structured and clearly annotated interface software   For example:  See Appendix 3 for an example illustrating different versions of a code fragment.   * using data sheets or calculations to assist in choosing appropriate component types and values for the interface   For example:  From data sheet, diode is rated at 3V and 20mA.   * testing and debugging a functional model of the interface for the sensor-controlled detection device   For example:  *I found that when the sensor was facing into the room, the detection device worked perfectly, but when I turned it around so it was facing the windows, it did not work anymore. I fixed this by making a little shield for the light sensor so that sunlight could not land on it. I also found that the light sensor worked OK one day but not the next. This was because the brightness of the light in the room changed. I fixed this by shining an LED onto the sensor and hiding the sensor from any other light. This made the sensor-controlled detection device work with any lighting conditions (see my annotated photographs of this).*  *The above expected learner responses are indicative only and relate to just part of what is required.* |

Final grades will be decided using professional judgment based on an examination of the evidence provided against the criteria in the Achievement Standard. Judgements should be holistic, rather than based on a checklist approach.

### Appendix 1: Learner evidence for Achieved

The code is all jammed onto consecutive lines, which does little to make the structure clear (even though it is essentially the same code as in the Excellence example). Labels are not used, and comments are non-existent. Even the descriptor at the start does not really capture the essence of what the barrier arm does. The code, while correct, is not that straightforward or easy to follow.

/\* This programme raises a barrier arm when a car is detected. \*/

int sensorValue = 0;

int Switch\_value = 0;

void setup()

pinMode(13, OUTPUT);

pinMode (0, INPUT);

pinMode (8, INPUT);

}

void loop() {

while (Switch\_value = = LOW)

Switch\_value = analogRead(8);

digitalWrite(13, HIGH);

while (Sensor\_value < 500)

Sensor\_value = analogRead(0);

digitalWrite(13, LOW);

}

### Appendix 2: Learner evidence for Merit

This piece of code is well structured, but not clearly annotated. Annotations are there, but they are not helpful in determining what tasks the code is addressing. .The comprehensibility of the code is compromised by a failure to use labels that give an intuitive understanding of the functions of the input and output pins on the microcontroller.

The comments highlighted in yellow are not part of what a learner might write. They are comments explaining where the annotation is incomplete.

/\* This piece of code waits until a switch is pressed. It then turns on a motor to raise a barrier arm.

A light sensor detects when the barrier arm reaches the ‘up’ position and the motor stops automatically.\*/

int arm\_up = 13; //create expression ‘arm\_up’ to refer to pin 13 where motor is connected

int sensor = 0; // Create expression ‘sensor’ to refer to pin 0 where sensor is connected.

int Switch = 8; //Create expression ‘Switch’ to digital pin 8 where switch is connected

int sensorValue = 0; // variable to store the value coming from the voltage divider

int Switch\_value = 0;

void setup() {

pinMode(13, OUTPUT); // declare the pin (13) as an output but no indication of what pin 13 //does

pinMode (0, INPUT); //Declare pin 0 as an analogue input.

pinMode (8, INPUT); //Declare the pin as an input, but no indication what its for

}

void loop() {

while (Switch\_value = = LOW) //

Switch\_value = analogRead(8); // read pin 8 but no indication what information this provides.

digitalWrite(13, HIGH); //Write pin 13 high, but no indication what this is doing.

while (Sensor\_value < 500) //while light beam not blocked

Sensor\_value = analogRead(0); //just just keep reading the sensor pin

digitalWrite(13, LOW); //Write pin 13 LOW, but no indication what this controls.

}

### Appendix 3: Learner evidence for Excellence

This piece of code is clearly written, well annotated, and well structured.

Yellow: Examples of given programme structures; others include “if” and “if-else” statements

Green: Examples of creating labels that make the code more readily understood

/\* This piece of code waits until a switch is pressed. It then turns on a motor to raise a barrier arm.

A light sensor detects when the barrier arm reaches the ‘up’ position and the motor stops automatically.\*/

int arm\_up = 13; //create expression ‘arm\_up’ to refer to pin 13 where motor is connected

int sensor = 0; // Create expression ‘sensor’ to refer to pin 0 where sensor is connected.

int Switch = 8; //Create expression ‘Switch’ to digital pin 8 where switch is //connected

int Sensor\_value = 0; // variable to store the value coming from the voltage divider

int Switch\_value = 0;

void setup() {

pinMode(arm\_up, OUTPUT); // declare the arm\_up pin (13) as an output

pinMode (sensor, INPUT); //Declare the sensor pin as an analogue input.

pinMode (Switch, INPUT); //Declare the switch pin as a digital input.

}

void loop() {

while (Switch\_value == LOW) //while light beam not blocked

Switch\_value = analogRead(Switch); // keep reading the switch pin:

digitalWrite(arm\_up, HIGH); //Once beam is blocked, start raising arm

while (Sensor\_value < 500) //while light beam not blocked

Sensor\_value = analogRead(sensor); //just just keep reading the sensor pin

digitalWrite(arm\_up, LOW); //once beam is blocked, stop arm from rising

}