

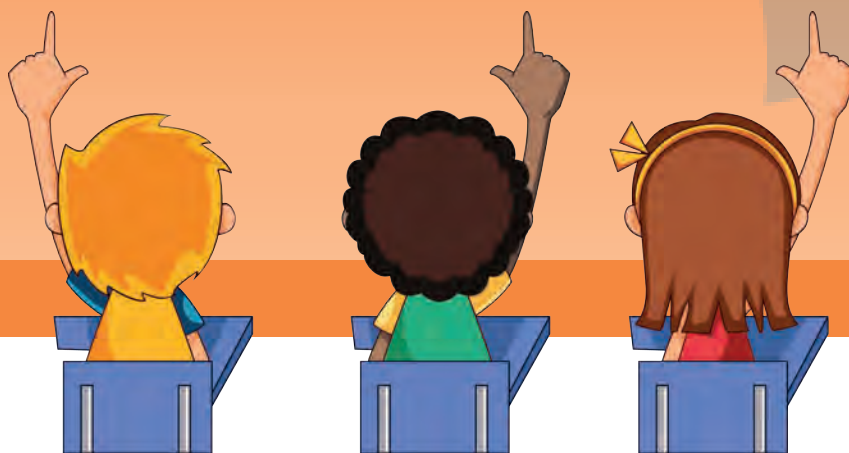
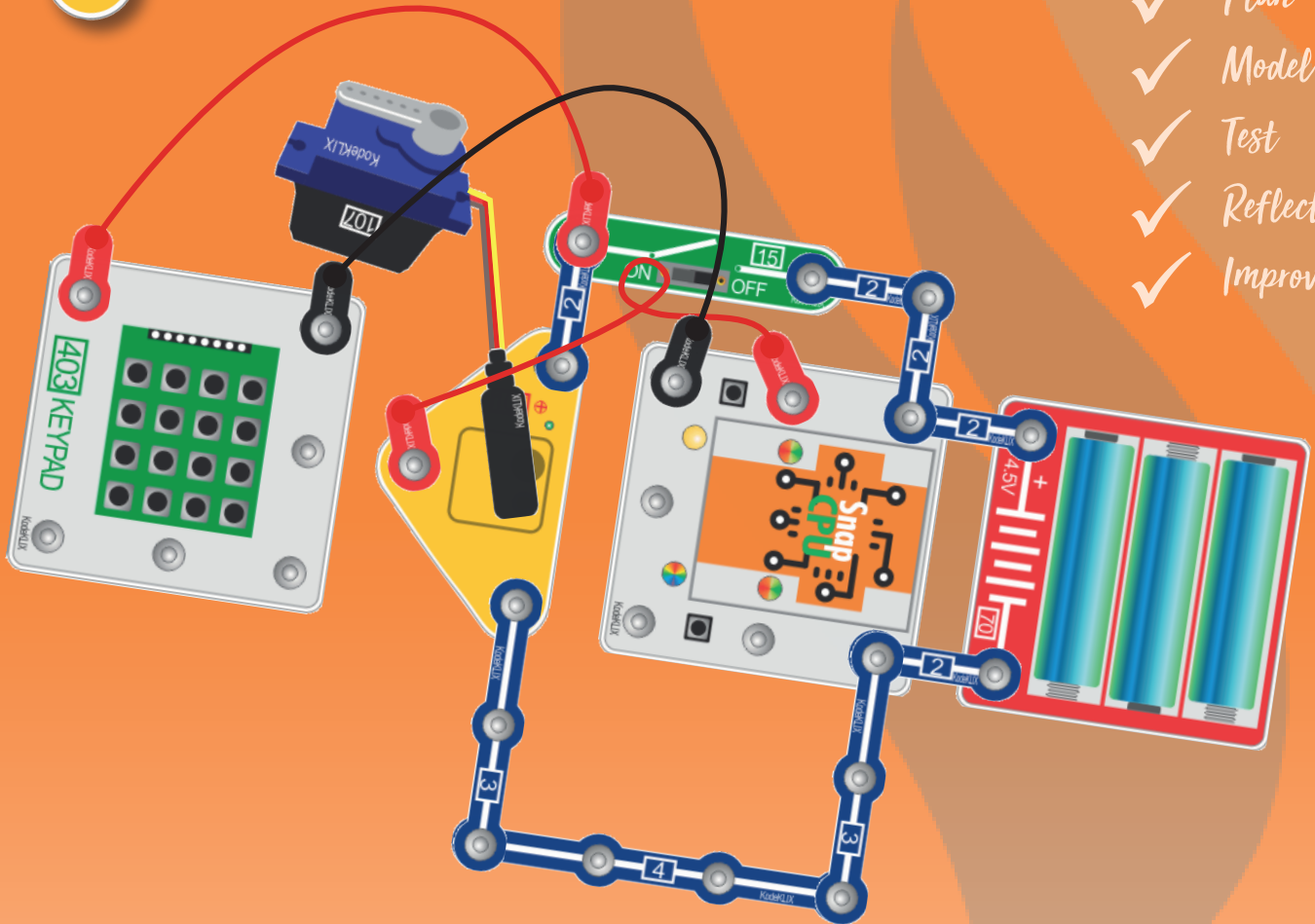
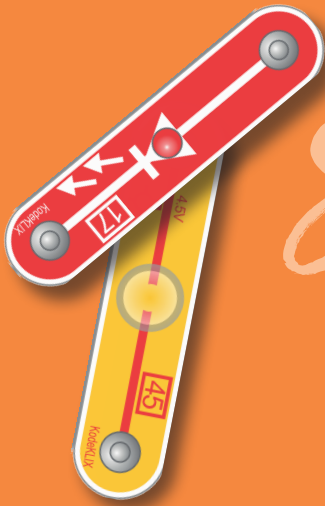
EXTENSION ACTIVITIES

STEM in Practice

with KodeKLIX®

- ✓ Define
- ✓ Plan
- ✓ Model
- ✓ Test
- ✓ Reflect
- ✓ Improve

AISWA SAMPLE



E

EXTENSION ACTIVITIES

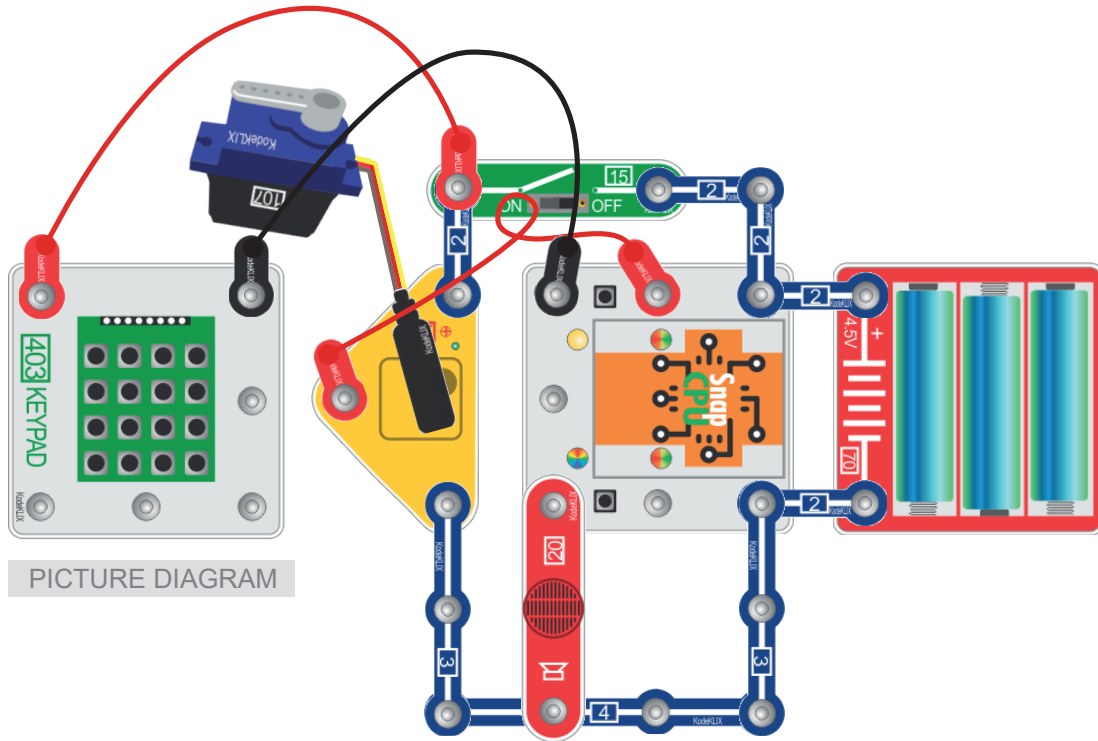
Expanding and experimenting on the concepts introduced in the Core Activities

| | | |
|----|---|---------|
| 1a | Controlling a servo motor: keypad | Page 6 |
| 1b | Controlling a servo motor: keypad and sound | Page 8 |
| 1c | Controlling a servo motor: multiple keypad buttons | Page 10 |
| 1d | Controlling a servo motor: light sensor (LDR) and sound | Page 11 |
| | | |
| 2a | Controlling motor direction: two switches | Page 13 |
| 2b | Controlling motor speed: KodeKLIX® speed block | Page 15 |
| 3a | Controlling motor speed: variable resistor (slider) | Page 16 |
| 3b | Controlling motor speed and direction: variable resistor (slider) | Page 18 |
| | | |
| 4a | Ultrasonic sensor: sensing objects | Page 19 |
| 4b | Ultrasonic sensor: sounder | Page 21 |
| | | |
| 5a | Microphone sensor: sensing sound | Page 23 |
| 5b | Microphone sensor: flashing LED | Page 25 |
| | | |
| 6a | Display matrix: bitmap display | Page 27 |
| 6b | Display matrix: bitmap and switch | Page 29 |
| 6c | Display matrix: text screen | Page 31 |
| 6d | Display matrix: two screens | Page 32 |
| | | |
| 7a | RFID sensor (Radio Frequency Identification): sound for tag | Page 34 |
| 7b | RFID sensor: servo motor | Page 36 |

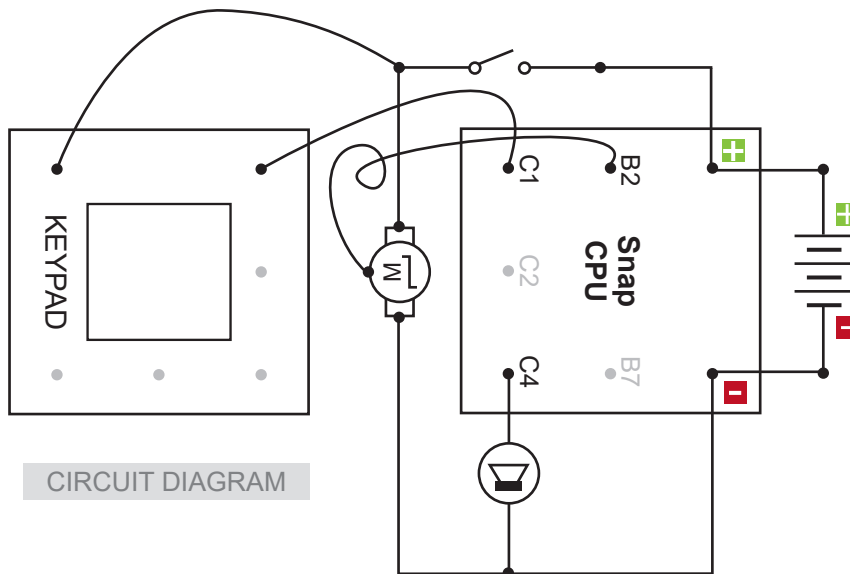
CODING

- 2
5
- 3
2
- 4
1
- 15
1
- 20
1
- 70
1
- 107
1
- 403
1
- CPU
1
- Servo Motor
- Speaker

B BASIC CIRCUIT COMPONENTS



PICTURE DIAGRAM



CIRCUIT DIAGRAM

MAIN IDEA

This circuit and code builds on extension activity 1a. A speaker is added to the circuit to indicate when the servo motor has moved. This would be helpful for example, if the circuit was designed to be a lock for a door and the sound would indicate the door lock is in place.

The circuit

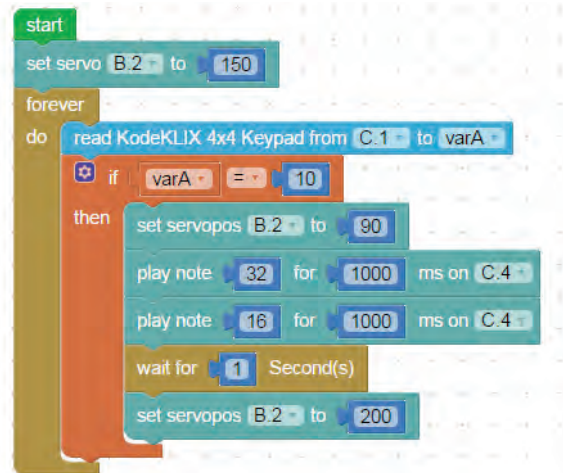
1. The SnapCPU requires power from batteries to work and a 'master switch' for circuit protection.
2. The **SERVO POT** (107) has one of its pins connected to pin B2 (B6) on the SnapCPU by a flying lead.
3. **KEYPAD** is connected to C1 with flying leads and **SPEAKER** is connected to C4.
4. Build the circuit. Make sure the master switch is turned **off**.

The code

1. The code is based on a loop command, multiple variable settings and an **if, then, else** decision command block. The loop will run forever. The number value you add will set the position of the servo motor. The number value you add for **play note** will set the note and duration from the speaker.
2. Two commands control the servo motor. Set servo initialises (wakes up) the servo motor. Set servopos moves the motor to a position when keypad number 10 is pressed and a note is played.
3. Write the code. The first set servopos value is 90 and the second set servopos value is 200.
4. Simulate the code before downloading to the SnapCPU.

Test the circuit with the code

1. Download the code to the SnapCPU.
2. Turn the master switch to **on** and observe the servo motor.
3. Press and then release number 10 on the keypad.
4. The servo motor should repeatedly move to each position with a sound played at the first set servopos.
5. If the circuit does **not** work, check your circuit very carefully. Are the power connections correct? Are any components in the wrong place or wrong way around? If the circuit **still** does not work, ask your teacher for help.



EXPERIMENTING

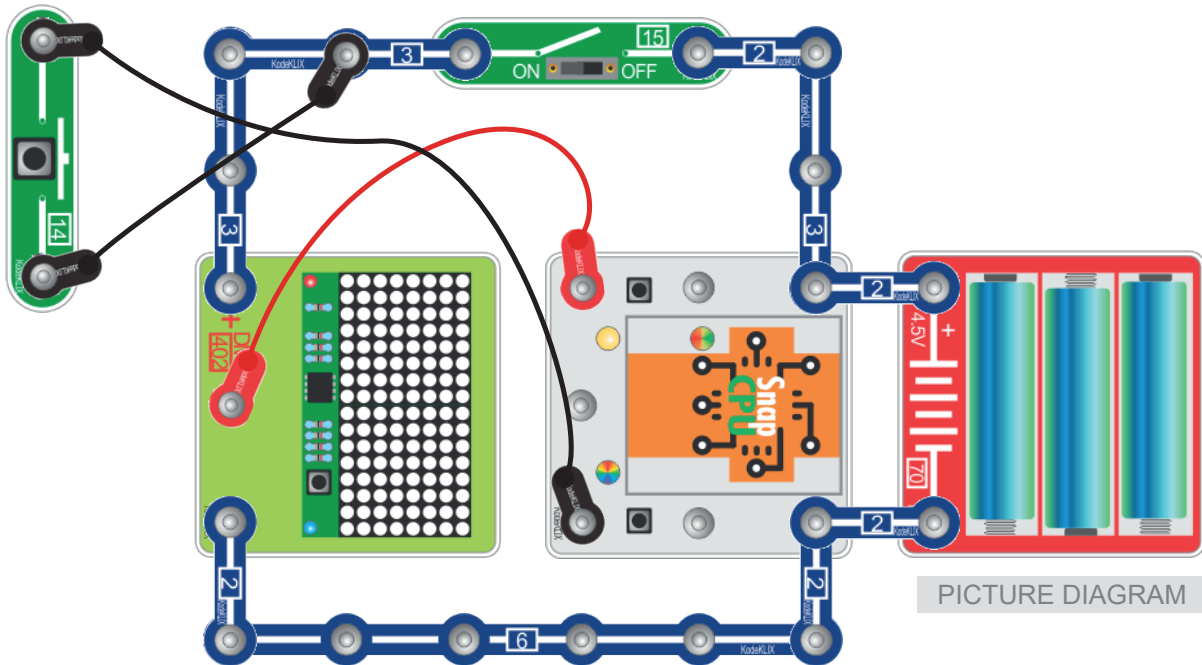
1. Experiment with the servo motor movement by changing the VALUE of the VARIABLE value for the keypad and servo position.
2. Experiment with the speaker sounds by changing the value of the note and duration of the sound.
3. Can you add speaker sounds for both servo motor positions?

Note: You will need to download the code with new variable setting every time a change is made, to make the servo motor move to a new position.

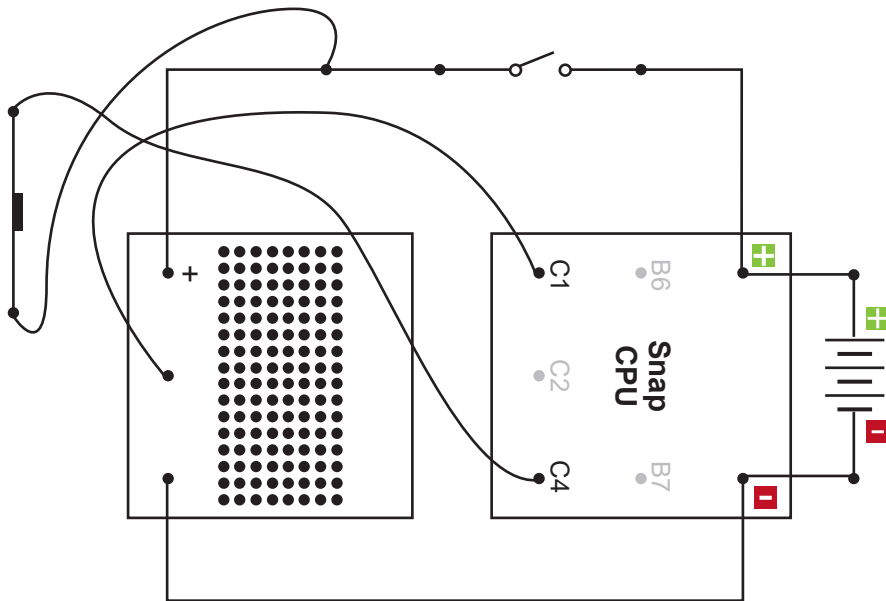
CODING

- 2
5
- 3
3
- 6
1
- 14
1
- 15
1
- 402
1
- 70
1
- CPU
1
-
-

D BASIC CIRCUIT COMPONENTS



PICTURE DIAGRAM



CIRCUIT DIAGRAM

MAIN IDEA

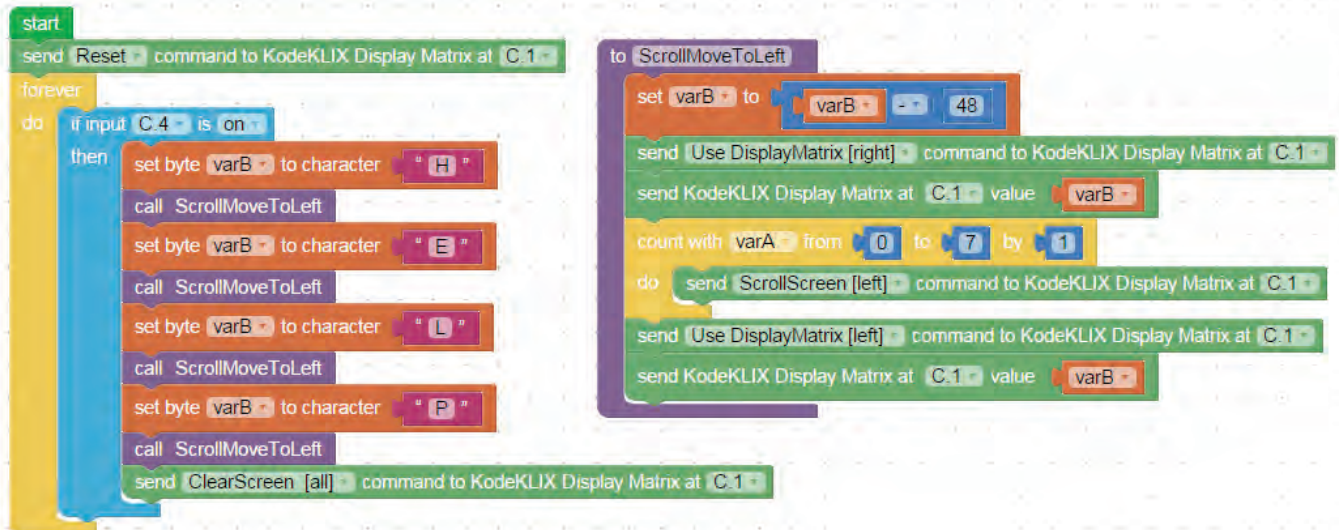
This circuit and code builds on extension activity 6c. The circuit uses a dual display screen (402). Coding for the circuit uses the send KodeKLIX® display text command block to display text across both screens. A switch provides an input for both display matrix screens. When the switch is pressed, information will be sent to the screens by using the send KodeKLIX® display for each letter.

The code

1. The code is based on a loop, **if, then** decision block a procedure **ScrollMoveToLeft** and set byte varB to the character (letter) you want to display. **If** the switch is pressed the information will be displayed. When the word has been displayed the screen is cleared using the **send ClearScreen [all]** command from the dropdown box.
2. The loop will continue running forever.
3. Each letter is sent individually to the display and moving from left to right across both screens starting in the right screen (HELP).
4. Write the code.
5. Simulate the code before downloading to the SnapCPU.

Test the circuit with the code

1. Download the code to the SnapCPU.
2. Turn the master switch to **on**, and observe the screen.
3. The word HELP will not display until the switch is pressed.
4. A display should appear similar to the ticks made on the set display matrix block.
5. If the circuit does **not** work, check your circuit very carefully. Are the power connections correct? Are any components in the wrong place or wrong way around? If the circuit **still** does not work, ask your teacher for help.



EXPERIMENTING

Experiment with the display matrix by using another input. Create new words by using different set byte characters.

Note: You will need to download the code with new variable setting every time a change is made.