Robu Creato 2.0

WARRIOR WITHIN

Covid Combat Bot

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

TEAM MEMBERS

LAKSHYA MITTAL
TARANG KAMBLE
ROHIT V. SAGVEKAR
KORADA PAVAN KUMAR

Contents:

1. Problem:

COVID-19 has made a huge impact on our lives and has made us realize how important it has become to reduce contact to ensure the safety of us and our loved ones. We have come a long way since covid has impacted our lives and developed various technologies to combat the impact of the virus. There is a clear need to devise more and more technologies to reduce the spread of the virus

2. Solution:

- We have come up with an innovative solution to go contactless in hospitals that do not require any sort of **human intervention**.
- It is an effective option to **reduce manpower** and at the same time ensure protection from COVID-19.
- Our unique bot is made to deliver **medicines** by making a note of the status of the patient's health and by taking vitals like the temperature of the body.

3. Description of our bot:

- This is a mobile autonomous bot that caters to the needs of COVID-19 medical patients with **minimal human contact**.
- It can distribute **medicines**, take vitals like the **temperature** of the body, collect queries from patients, etc., all these are implemented with no human interference.
- It can also **sanitize hands** and **deliver water** to the patients at regular intervals.
- This will help in creating a safe environment for **frontline workers.**
  .
- This bot can be used for other services too where there is a **queue system**, for example serving in restaurants.

4. Sensors and modules:

- **MLX90614** - Contactless temperature sensing of a person.
- **IR sensors** - It will help identify the paths.
- **LCD**- For displaying the instructions.
- **Ultrasonic sensor**- For detecting the obstacle.
- **Arduino-Mega** - It will be used as a microcontroller.

- **Motor**- Rotating the wheels of the bot and opening and closing the doors using a lead screw mechanism.
- **Motor Driver**- It will control the motors.
- **Sanitizer Dispensers**.

5. Advantages:

- It can work even on curved paths.
- It is a cost efficient bot which costs around ₹2500.
- It reduces human effort.
- This will reduce person to person contact.

6. Appearance of the bot:

- It is basically a **shelf** that can traverse from one bed to another.
- The **multi-compartment** functionality will help in segregating the medicines for different beds.
- The particular compartment will **light** up as soon as the bed reaches that bed.
- It is having an **LCD screen** that will **instruct** the patients.
- The upper portion of the bot consists of an **IR thermometer** that will measure the temperature without contact.
- It will also have sanitizer and other equipment as well.

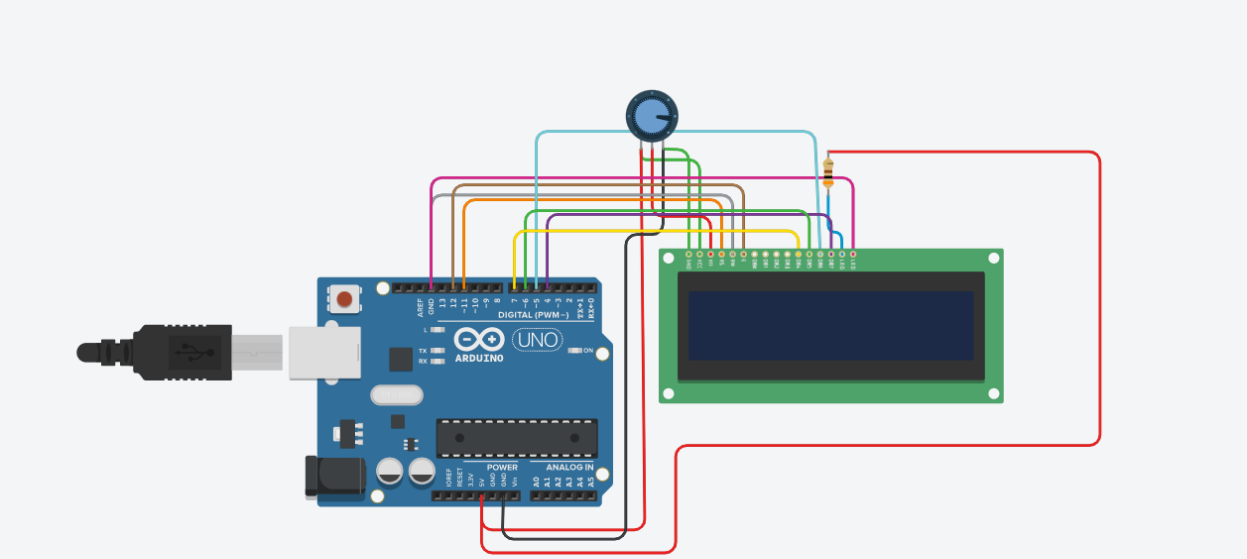
7. Technicalities of the bot:

- It follows the **LSRB algorithm** to **automatically** detect the nodes or the junctions along the path that it has to follow.
- It follows the path to reach the designated place to deliver the needful.
- It is incorporated with an **obstacle detection mechanism** which is integrated with an **interrupt service routine**.
- The **infrared sensors** play a very important role for the bot to figure out the junctions and move according to a given set of instructions.
- We will be using a **keypad** to give instructions to our bot from the base station.

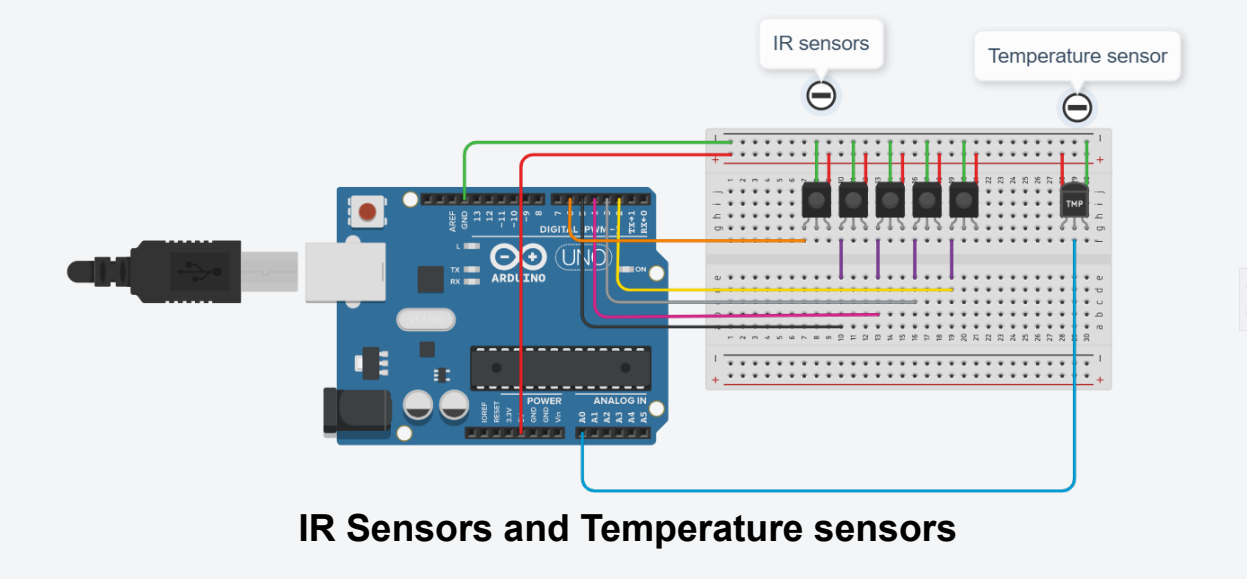
- After the bot reaches its desired location, It will perform the following tasks:
    - Delivering the **medicines** and lighting up that compartment.
    - Showing an **LCD screen** to give instructions
    - Measuring the **temperature**.
    - **Sanitizing** the hands, **delivering water**.

- The drawers will open automatically and will close as soon as the medicines are collected.

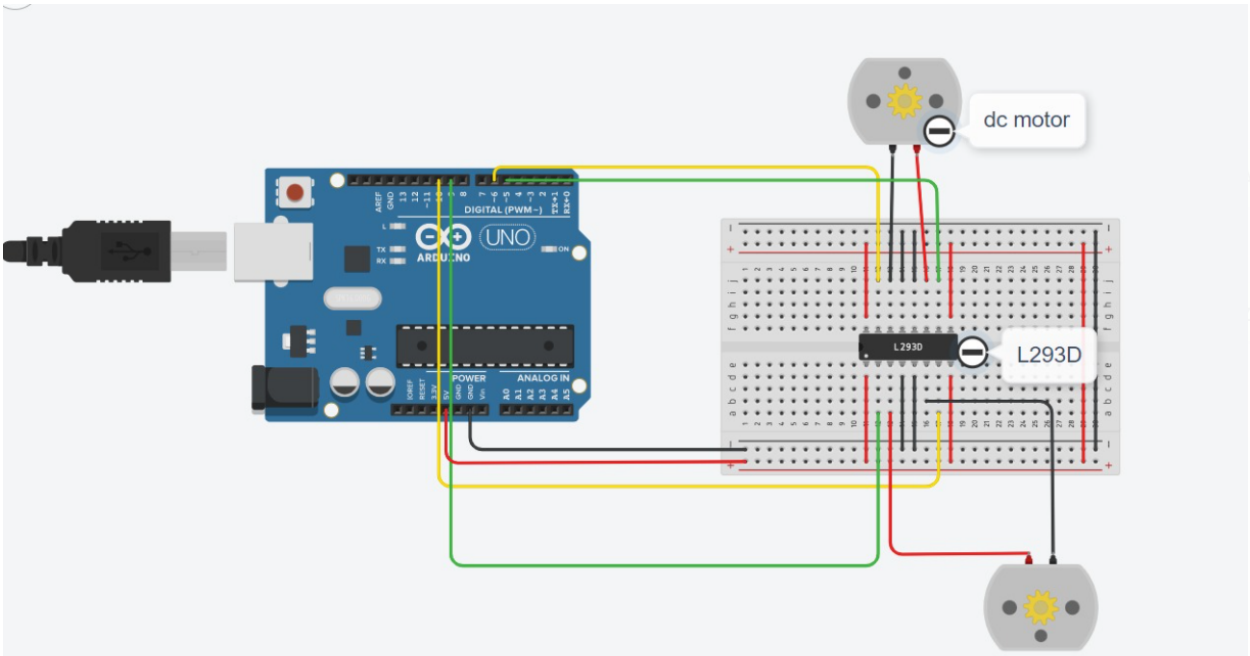
8. Circuit diagram of the sensors once the bot reaches particular bed:



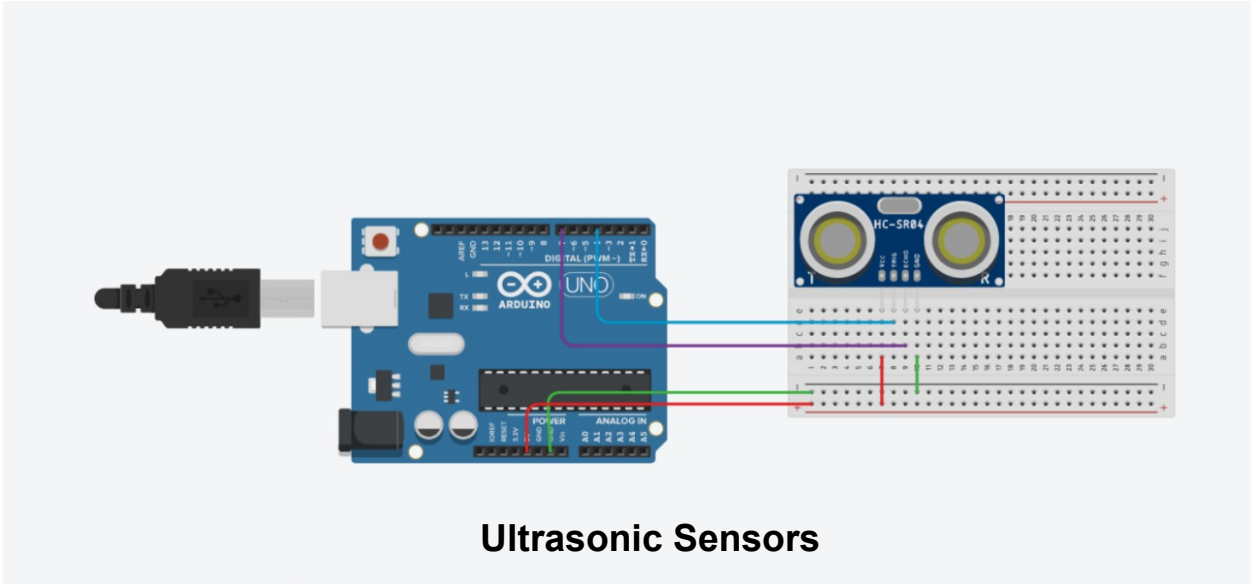
LCD Display



IR Sensors and Temperature sensors

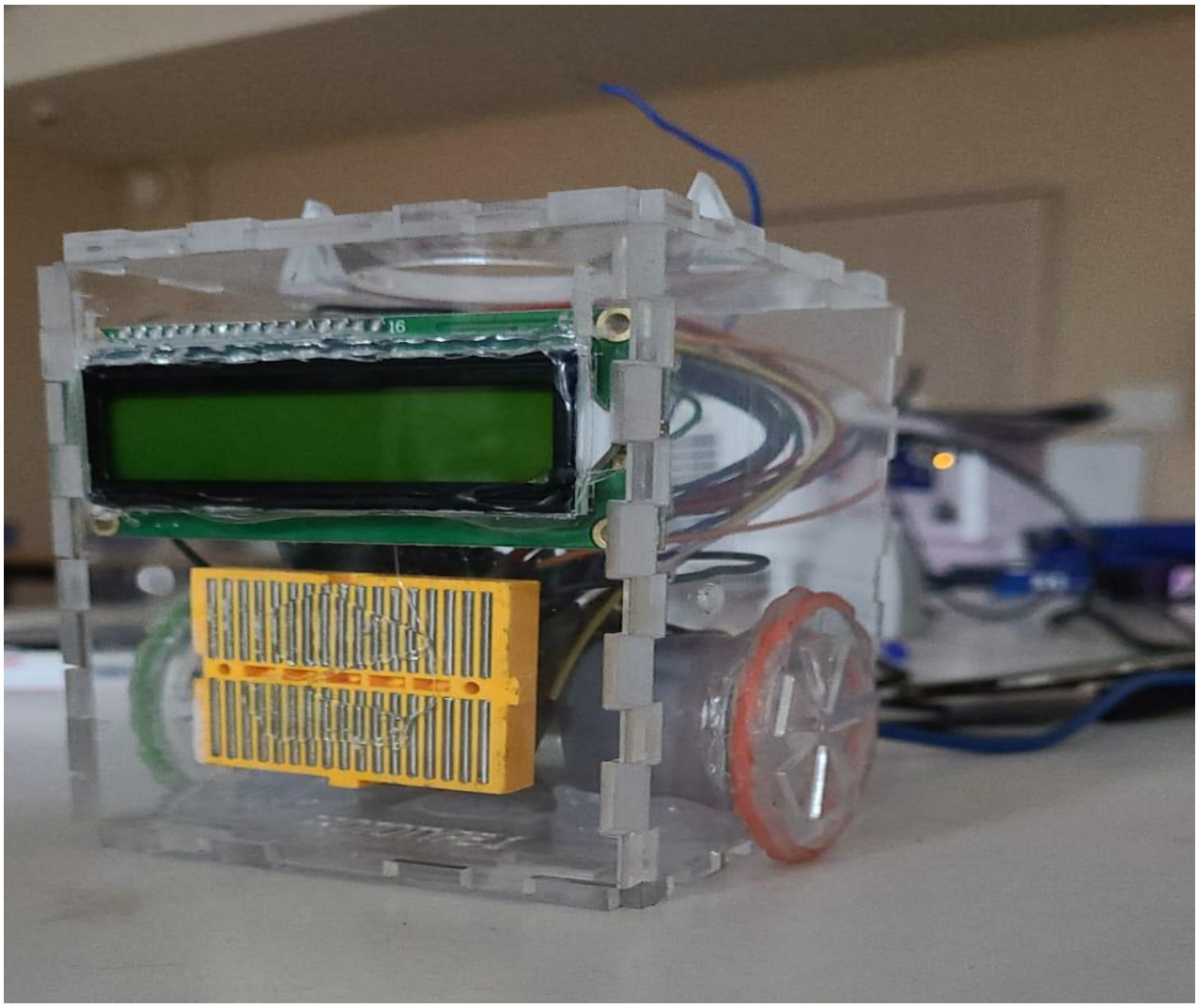


Motors and Motor Driver



Ultrasonic Sensors

***All of the above circuits will be made on only one Arduino Mega and are only shown on different Arduino UNOs to show the circuits clearly**



This is the first prototype of our bot which has motors, motor driver, LCD and Arduino Mega. Due to lack of hardware resources, we have created this mini model to verify our algorithm in the limited timespan.

9. Market Analysis:

- The health sector of India is as big as any sector of any country including both 43486 private hospitals and 25778 public hospitals. This covid bot can be implemented on both scales as this is financially feasible to implement.

- The food corporations and junk 4000 food companies and their franchises like dominoes and burger king etc will have greater usage and level of implementation more suitable.

- MVP cost = 1500 Indian rupees + maintenance charge that can be neglected
  1500* (the percentage of hospitals which can afford such bots * total- product existing in the market)
  Percentage of hospital = would be tending to about 70 percent, as implementation in public hospitals depends on government and private hospitals can afford this much amount easily.

1500* 43000(total private hospitals)= 6.4 crore
Profit can be calculated accordingly.

- As there are multiple areas where this bot can be used but primarily targeted personas are the medical sector, further advanced features can be added according to requirements and cost will increase according to that.

- And if the government wants to implement in the public hospitals, it would be beneficial for the social sector as well

10. Target Users:

- The Bot can be used in multi-bedded hospitals where we want to minimize the contact and hence prevent infection.

- It can be used in restaurants to ensure contactless delivery of food.

- Can also be implemented in shopping complexes so that the shopped items can be kept in one place.

11. Future enhancements:

- We can give inputs to the bot through NodeMcu which can send messages in the form of a proper schedule with dates and timings.

- The schedule is set for particular beds so that it can do the needful at predefined intervals of time.

- We can add more testing features to it e.g. make a talking bot.

- Wireless docking station for the bot to automatically charge itself.

12. Pseudocode

Initialize 2 dimensional array bedMap which'll store the address of each bed from terminus/docking position

Declare a function which returns an integer which represents length of array and takes input as an array
```
FOR i = 0 to 50
    IF the i'th value in the input array is 'A'
        RETURN i
```

Initialize 1 dimensional array 'presentLocation' which'll store the current location of bot from terminus/docking position

Initialize 1 dimensional array 'tempAddress' which'll denote the turns the robot has to take to reach next bed

initialize 1 dimensional array 'targetLocations' which'll hold the bed numbers which the bot has to visit

initialize 'numberTarget' which is length of array 'targetLocations'
    initialize 'currentTurn' which is the number of turns the robot has taken while going from one bed to another
Initialize 'currentBed' which is the number of the bed it just left
initialize 'Bn' (BedNumber), the number of beds it has visited
Initialize 'Tn' (TurnNumber), the number of turns it has taken

//Declaring the pin numbers
Declare pin 3 as 1st pin of 1st motor
Declare pin 2 as 2nd pin of 1st motor
Declare pin 4 as 1st pin of 2nd motor
Declare pin 5 as 2nd pin of 2nd motor
Declare pin 6 as data input of 1st IR Sensor
Declare pin 7 as data input of 2nd IR Sensor
Declare pin 8 as data input of 3rd IR Sensor
Declare pin 9 as data input of 4th IR Sensor
Declare pin 10 as data input of 5th IR Sensor
Declare pin as 1st pin of drawer motor
Declare pin as 2nd pin of drawer motor

Declare pins 11, 12, 13, 14, 15, 16 as pins for the LCD display

Initialize string variable screen_print1 which'll show the first message
Initialize string variable screen_print2 which'll show the second message

Initialize pin for LED
initialize pin for temperature
Initialize integer variable 'temp' for temperature
initialize integer variable 'dur' for duration that the drawer will be open

Initialize five integer variables to store readings of the 5 IR sensors
initialize five integer variables to store the previous readings of IR sensors

initialize integer variable 'stopTime' which is the time between reading IR sensor values
initialize integer variable 'turnTime' which is time taken to take a right turn
initialize integer variable 'nodeTime'

//Initializing the various flags in this section
initialize flag 'stationFlag' which'll be raised when bot reaches a station/bed
initialize flag 'terminusFlag' which'll be raised when bot reaches the terminus/docking position
initialize flag 'mapFlag' which'll be raised when bot is in mapping mode

initialize flag 'moveFlag' which'll be raised when bot is in moving mode
initialize flag 'xFlag' which'll be raised when the bot encounters a four-way junction
initialize flag 'tStraightFlag' which'll be raised when the bot encounters a straight 3 way junction
initialize flag 'tLeftFlag' which'll be raised when the bot encounters a leftward 3 way junction
initialize flag 'tRightFlag' which'll be raised when the bot encounters a rightward 3 way junction
initialize flag 'lLeftFlag' which'll be raised when the bot encounters a leftward 2 way junction
initialize flag 'lRightFlag' which'll be raised when the bot encounters a rightward 2 junction
initialize flag 'shortpathFlag' which'll be raised when bot needs to shorten the address in tempAddress

initialize function turnLeft() which'll make bot turn left 90 degrees
initialize function turnRight() which'll make bot turn right 90 degrees
initialize function turnStraight() which'll make bot move ahead from a junction
initialize function turnBack() which'll make bot turn back 180 degrees and move a little forward
initialize function goAhead() which'll ensure robot stays on track with the path always between IR sensors 2 and 4
initialize function goBack() which'll cause the bot to reverse
initialize function straightCheck() which'll move the bot a little forward to differentiate between station, junction and terminus
initialize function backCheck() to come back after straightCheck

initialize function detectNode() which'll classify the type of junction the robot is at
initialize function shortPath() which'll make sure the previous path is cleared so that each row of bedMap is the address to that bed from the terminus
initialize function screen_print() which'll print given string on the LCD display
initialize function Drawer_out() which'll open the drawer for given number of milliseconds
initialize function Drawer_in() which'll close the drawer for given number of milliseconds
initialize function ledblink() which'll make the indicator LED blink

VOID setup(), the setup loop
  initialize Motorpins, led as OUTPUT
  initialize Sensor pins, thermometer as input

FOR i = 1 to 50
  FOR j = 1 to 50
    Assign the letter 'A' to all places in bedMap

VOID LOOP() which will continue to run indefintely when arduino is turned on
  CALL readSensor() to read the input from the IR sensors
  IF mapFlag is raised
    goAhead following the road for stopTime milliseconds
    CALL detectNode to check if bot is at a junction
      IF the node allows a left turn
        Turn left
        store 'L' in the Tn'th position of Bn'th row of bedMap
        INCREMENT Tn by 1

```
            IF shortpathFlag is raised
               CALL shortpath()
         ELSE IF the node doesn't allow a left turn
            Go straight
            store 'S' in the Tn'th position of Bn'th row of bedMap
            INCREMENT Tn by 1
            IF shortpathFlag is raised
               CALL shortpath()
         ELSE IF the terminusFlag is raised
            Clear bedMap matrix
            Reset Tn to 0
            Lower mapFlag
            CALL turnBack()
         ELSE IF the stationFlag is raised
            CALL turnBack()
            INCREMENT Bn by 1
            COPY Bn'th entry to the Bn+1st entry of bedMap
            INCREMENT Tn by 1
            IF last entry of bedMap is 'B'
               RAISE shortpathFlag


   IF moveFlag is raised
      CALL goAhead()
      CALL detectNode()

      IF it's the first bed
         WHILE the entries in bedMap are a
            COPY the address of the first bed in targetLocations to tempAddress
      ELSE IF it's not the first bed
         Reverse the entries of current address and copy those into tempAdress, add the rest of the entries from
last commmon node to target bed


      IF a junction is detected
         Take turn based on the tempAdress matrix

      IF a station/bed is reached (Using idea of finite state machine)
         PRINT the first message on LCD screen
         PUSH out the drawer
         TURN ON the led
         WAIT for 500 milliseconds
         PRINT the second message on LCD screen
         PULL in the drawer
         TAKE temperature reading
         INCREMENT currentBed by 1
         RESET currentTurn
```

LOWER all flags.