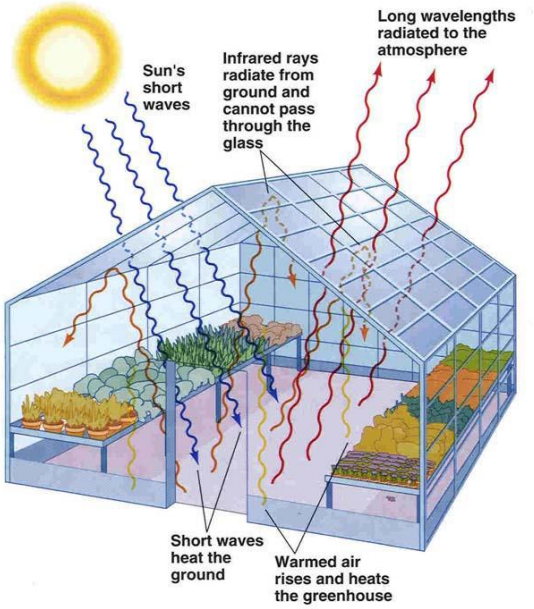
Greenhouse project

In this project all the elements from previous lessons come together as you must integrate it into a simple Greenhouse controller.

The Greenhouse controller must read input from different sensors attached to the Beaglebone Black and present it on a webpage, which also has controls to activate some actuators.



Sensor parameters:

- Light intensity
- Temperature
- Humidity

Actuator parameters:

- LED lighting with variable intensity
- Window open/close
- Heater (represented by a LED)

You can use the transparent plastic box as the greenhouse and mount a servo with double-sided tape so the arm on the servo can swing the lid open or closed.

Requirements for the system:

- All user interaction must be via a web interface
- Administrator action must be possible remotely, via an ssh-connection
- User must be able to read the temperature in the greenhouse
- User must be able to read the humidity in the greenhouse
- User must be able to read the light intensity in the greenhouse
- User must be able to see window position (open/closed)
- User must be able to see heater status (on/off)
- User must be able to set a light intensity level in the greenhouse
- User must be able to open and close the window in the greenhouse
- User must be able to activate/deactivate the heater in the greenhouse

Requirements for the documentation:

- Header in all files with filename, author and date
  - Header above all functions/methods with name, purpose, description of input parameters and return values.
- Document your code with plenty of comments
- Document which Beaglebone pins are used for which sensors and actuators (e.g. in a table).
- VIA standard report layout
- Document your interface circuits (include block and circuit diagrams)

- Document your code (include relevant UML diagrams, doxygen reports, etc.)
- User manual in appendix

Deadline:

Announced on itsLearning.

Hint:

The user interface can be made as a node script that instantiates a web server. External binary programs or shell scripts can be called from the script, and their output send to the browser via the socket.io api. See the *timeserver* example (get_bbb_time.zip) that was uploaded, on itsLearning, in an earlier session. In this way you can reuse your C-programs for setting light intensity and read temperature/humidity sensor and let a node script handle the user interface.

ERL 17/11/2020