

# Smart Plant Monitoring System

Sreeram Sadasivam   Vishwanath Vadhri   Christoph Peusens  
Supradha Ramesh   Mallikarjun Nutti  
FB 20 Informatik  
Technische Universität Darmstadt  
*{sreeram.sadasivam,viswanath.vadhri,christoph.peusens,  
supradha.ramesh,mallikarjun.nuti}@stud.tu-darmstadt.de*

## Abstract

*Plant monitoring is seen as one of the most important tasks in any farming or agriculture based environment. With the inception of Ambient Intelligent systems, there have been a rise in ambient intelligent based devices - Smart Homes[1] and other similar technologies involving RFID has evolved over the past few years[2]. Integration of such an ambient intelligent system with plant monitoring makes farming easier. In this paper, we discuss about the implementation of a smart plant monitoring system which makes use of the concept ambient intelligence with the use of .Net Gadgeteer which, proactively handles the plant monitoring system. The given implementation works along with a cloud based server and a mobile based device (ideally Android/iOS device) which helps the user to control and see the status of the plant which is being monitored by the hardware device. The given circuitry detects changes in the moisture, temperature and light conditions in and around the plant, and performs a machine based curation on the plant by providing necessary irrigation and illumination for the plant. Machine curation is also integrated with active weather forecasting systems which are deployed in the cloud based server using which advanced machine curation is performed. For user based curation, the Android device provides user an option to override a machine curated operation.*

## Motivation

The most important factors for the quality and productivity of plant growth are temperature, humidity, light and the level of the carbon dioxide. Continuous monitoring of these environmental variables gives information to the grower to better understand, how each factor affects growth and how to manage maximal

growth of plants. Climate control and monitoring of the plant is one of important aspects in agriculture. The aspects we are presenting resembles the concept of precision agriculture, this is the trend of farming presented in recent years for commercial and research agriculture. In the precision agriculture framework the focus is mainly on understanding the environment through the interpretation of a wide variety of data coming from GPS systems, satellite imaging, and in-field sensors. Cypress is one of top semiconductor company in the world. Currently they have envisioned the scope by moving towards Internet of Things, with the major focus on “Internet of plants”. The main motivation of the project is for the user to monitor the plants or cultivation to get enough resources such as light and water without the user need to be present at the plants or cultivation area, and also could manipulate the resources provided to the plants depending on the climate of the plant’s location(eg : if there is enough rain or moisture for the day, the sensor in the soil would detect accordingly for the operation to begin and intimate the user). This could help user not only to give the resources to the plants everyday without much manual effort also helps the constant and healthy growth of a plant.

## Related Works

Mancuso et al [3], has done a similar research work in a tomato greenhouse in the South of Italy. They are using Sensicast devices for the air temperature, relative humidity and soil temperature measurements with wireless sensor network. They have also developed a Web-based plant monitoring application. Greenhouse grower can read the measurements over the Internet, and an alarm will be sent to his mobile phone by SMS or GPRS if some measurement variable changes rapidly. Bridge node gathers data from other sensor nodes, which transmit the measurements of temperature and relative humidity in one minute intervals.

Teemu Ahonen et al [4], has done the research in Martens Greenhouse Research Center’s greenhouse in the Närpiö town in Western Finland, they had integrated three commercial sensors with Sensinode’s sensor platform to measure four environmental key variables in greenhouse control. The system feasibility was verified in a simple star topology setup in a tomato greenhouse. The sensors used were SHT75 humidity/temperature sensor and TSL262R light irradiance sensor, and Figaro’s TGS4161 CO2 sensor used. Application of the concept in the greenhouse: temperature, luminosity and humidity sensors measured climate variables and communicated directly with the gateway node. The gateway node acted as a coordinator and received the measured data from the sensor nodes. The maximal communication range, 15 meters was figured out in individual test where the distance between the coordinator and the sensor node inside the greenhouse dense flora was increased, the reliable communication range fell to one third in the greenhouse’s dense flora.

## Design

The proposed system consists of two curation systems. The first one being the *machine based curation* and other being the *user based curation*. These curative systems are in place to provide a smart and involuntary feedback to the given environment. The former is more of predictive system devised on a hardware whereas, the latter is for the user to actively handle the response provided by hardware. The curation handled by the device are based on the sensor information it receives from its end points(*sensory hardware units*), such a curation is called as “*Direct Machine Based Curation*”. There is also a hybrid machine based curation provided by the device and the cloud server which relies on the Weather Forecast information with the collects sensor values from the endpoints, such a curation can be called as “*Advanced Machine Based Curation*”. If the user wants to take over the system then, it becomes more of a “*User Based Curation*”. On top of all, the proposed system design is a pub-sub system. The Monitoring hardware is the publisher and Mobile based devices are subscribers. The cloud based server act as “*brokers*”.

## Basic Interaction

Primarily the interaction between the monitoring device and the user enabled mobile device happens as a publisher-subscriber system. This interaction is brokered by the cloud based server with monitoring hardware and the mobile device as the publisher and subscriber respectively. The Cloud Based server - “Parse” <sup>1</sup> provides core functionalities. This server not only gives a hand for machine based curation but also, in providing various push based services to the Mobile Device(Android/iOS/Windows). The monitoring hardware used for this system is .Net Gadgeteer <sup>2</sup>. Before any curation to work the device must initially register to the cloud server by their respective device tokens thereby, the broker service can later on perform push services. For handling the registration operation, the server have provided two different registration APIs — one for the Android/iOS/Windows device using which mobile device can register itself and also subscribe to a particular monitoring device. And the other for the Monitoring hardware device in order for registering to server with its Geo-Location. Once the registration is successful the device and user can perform curations. The mobile application deployed in the mobile device informs the user about the device status and weather status in the location of subscribed monitoring device using which user can perform the curation of the system(*Ref:Table 1*).

API	Type	Target	Summary
-----	------	--------	---------

---

<sup>1</sup>See [www.parse.com/docs](http://www.parse.com/docs)

<sup>2</sup>See [www.netmf.com/gadgeteer/](http://www.netmf.com/gadgeteer/)

<i>register</i>	Registration	.Net Gadgeteer	Registers the Device ID of Gadgeteer to the server and also provides the Geo Location to the server.
<i>extDeviceRegister</i>	Registration	Android/iOS	Registers the mobile device to the server and subscribes to a gadgeteer device by providing the Device ID of Gadgeteer as one of the inputs.
<i>deviceStatus</i>	Device	Android/iOS	Provides the status such as water, light modules of monitoring device to which the mobile device has subscribed to.
<i>updateDeviceStatus</i>	Device	.Net Gadgeteer	Updates the device status of the monitoring hardware device in the server. When such a modification happens an event is triggered to all the subscribed mobile devices(a push based service).
<i>manualOverride</i>	Device	Android/iOS	Modifies the device status of the monitoring hardware device in the server. When such a modification happens an event is triggered to the subscribed gadgeteer device(a poll based service).
<i>pollForDevice</i>	Device	.NetGadgeteer	Polling mechanism implemented for monitoring hardware thereby, the hardware can poll for events created by mobile interfaced user.

<i>weatherStatusForDevice</i>	Weather	Android/iOS	Provides the weather condition in the location of monitoring device to which the mobile device has subscribed to.
<i>weatherForecastForDevice</i>	Weather	Both	Provides the weather forecast to the gadgeteer for the timeline of closest 3 hour time slot. In case of mobile device, it provides an overall forecast which includes for the next 24 hrs from the current time.

Table 1: API Table

## Machine Based Curation

As said stated in main section, machine based curation works in two modes — *Direct* and *Advanced*. In case of “*Direct*”, the cloud server acts as a “*broker*” between the Mobile device and the Monitoring Hardware. Whereas in case of “*Advanced*”, it is more of a hybrid model where both weather forecast information provided by the server and sensor data collected by the device are considered before performing an event.

### Direct Mode

The sensor nodes detects the changes in the environment around the monitoring plant such as - light, moisture. Based on these values the monitoring device will curate the events to be handled in the system. The curation is very straight-forward due to which it is called as “*Direct*”. The monitoring hardware is integrated with moisture and light sensors which can help in providing valid curations to the system. Moisture sensors readings are compared with the threshold values set in the monitoring hardware and necessary actions are performed. If the moisture reading is low and, as per the plant configuration the watering for the plant is required then, the plant will get watered accordingly. The same happens with light illumination on the plant, the light sensor deployed with the Gadgeteer detects the light change and accordingly turns ON the artificial light provided. The direct mode completely relies on the sensor nodes and gadgeteer device.

### Advanced Mode

Hybrid machine curation is based on predictive inputs provided by the server and the sensor inputs from the sensory nodes. For the implementation of such a mode, the system requires to get access of weather forecast of the location

where the monitoring unit is present. Such an implementation is handled by the *weatherForecastForDevice* API provided by the server. Such a service is invoked by the monitoring device only when it believes it has encountered a minimum threshold for its activity. For example, the monitoring device when it detects the reduction in moisture content in system and requires the “*Direct*” mode to take over. Instead of such a takeover the “*Advanced*” mode of operation is initiated. In this mode of operation, the device performs curation with the server as well. Thus making the system more of a hybrid model. When the monitoring hardware decides to invoke the weather forecast API, it sends its current device time using which the server can send back the forecast time slot for the device. The time slot mechanism is handled based on the time provided by the device.

---

**Algorithm 1** Calculate Time Slot

---

```

timedif  $\leftarrow$  devtime – curtimeslot
if timedif  $\geq$  0 then
    timeslot  $\leftarrow$  timedif/timeslotduration
    if timeslot  $\leq$  maxtimeslot then
        if weatherinfo[timeslot]  $\neq$  “Rain” then
            isRain  $\leftarrow$  false
        else
            isRain  $\leftarrow$  true
        end if
    else
        return  $\leftarrow$  “InvalidDevicetime”
    end if
else
    return  $\leftarrow$  “UsePreviousForecastData”
end if
return  $\leftarrow$  isRain

```

---

## User Based Curation

The user based curation is hand held by the user using the Android/iOS devices. The user can curate the device and weather statuses of the monitored device using his/her mobile device. Using the data provided by the android application, the user can perform override functionalities (if required) by the user. The standard APIs provided by the server can be used by the user’s mobile application to handle the device and weather statuses. The user can also invoke the weather forecast API which was provided for advanced machine curation. The forecast API does not work the same way as for the gadgeteer. In case of invocation from the mobile device, the forecast API provides a generic weather forecast for the next 24 hrs. Using such an information the user can perform much more intelligent decisions. User curation is handled by having the option for the user

to perform manual override option from the Android device. Since the current implementation of the Parse SDK does not support the Gadgeteer. The system performs a polling service from monitoring hardware. The Gadgeteer polls the server to see if there is any new event waiting for the Gadgeteer to service upon. If it finds the servicing, then it handles the change and performs a update device status API call to the server. Thereby, triggering a push to the user's mobile device.

## Working with Modes

The Gadgeteer initially sets itself in “*Advanced*” mode and later on changes to “*Direct*” mode based on the correctness of the forecast. When the forecast fails the gadgeteer retracts itself to “*Direct*” mode. For more info refer Algorithm 2.

---

### Algorithm 2 Mode Change Over

---

```

mode ← “Advanced”
if forecast is success then
    return ← “Nomodechangeneeded”
else
    mode ← “Direct” until next time slot
end if
return ← mode

```

---

## Evaluation

The proposed system encounters multiple change-over schemes with respect to the curation mechanism in place.

---

### Scenario 1

---

Notification given by hardware device to user(weather forecasting with the chance of rain)

The device set in advanced mode.

As mentioned in the design and basic interactions the hardware device and android device acts as the pub-sub system and the cloud- based server acts as the broker between both systems. The system works as follows:

- The hardware device checks the moisture or humidity of the plant soil and if the threshold value( when the plant needs water ), the weather forecast has been invoked to check if in next 1 hour if it would rain or not.
  - If there is chance of rain, the event for watering the plants has not been triggered.
-

Scenario 2: Notification given by hardware device to user (weather forecasting with no chance of rain) The device set in advanced mode. As mentioned in the design and basic interactions the hardware device and android device acts as the pub-sub system and the cloud-based server acts as the broker between both systems. The system works as follows: The hardware device checks the moisture or humidity of the plant soil and if the threshold value (when the plant needs water), the weather forecast has been invoked to check if in next 1 hour if it would rain or not. If there is no chance of rain, it issues the event change to the user and starts watering the plants. If the user doesn't want the plants to be watered, he can override the event and stop from watering the plants. Now after the manual override the device mode is set to manual mode. In manual mode (for certain time "X" period, only the manual override would be set. As to avoid continuous notification and working of the hardware.

Scenario 3: Notification given from user to hardware device The device is set to manual mode. As mentioned in the design and basic interactions the hardware device and android device acts as the pub-sub system and the cloud-based server acts as the broker between both systems. The system works as follows: The user manipulates the resources to his plant according to the convenience of user. If the user is at different location from the location of the plant, the user can request for the (invoke) weather forecast of the plant location and according to the forecast the user can decide on what resources to be given to his/her plant.

Scenario 4 : Notification given by hardware device to user (light resource) The device is set to advanced mode. As mentioned in the design and basic interactions the hardware device and android device acts as the pub-sub system and the cloud-based server acts as the broker between both systems. The system works as follows: The hardware device checks the light threshold value and sends a notification to the user regarding its event for switching on the light. If the user wants to override the event, he could manipulate it as mentioned above.

## Inference

## Acknowledgment

The authors would like to thank Florian Müller of Tele-Kooperation(TK) from Informatik department of Technische Universität Darmstadt for providing the opportunity to work in the selected topic.

## References

- [1] Kidd, CoryD. et.al. *"The Aware Home: A Living Laboratory for Ubiquitous Computing Research"*. Springer Berlin Heidelberg (1999).
- [2] Antonio J. Jara, Miguel A. Zamora, and Antonio F. Skarmeta. *"An internet of things—based personal device for diabetes therapy management in ambient"*



*assisted living (AAL).*”. Personal Ubiquitous Comput. 15, 4 (April 2011), 431-440.

- [3] M. Mancuso and F. Bustaffa, “*TA Wireless Sensors Network for Monitoring Environmental Variables in a Tomato Greenhouse*”.
- [4] Teemu Ahonen, Reino Virrankoski and Mohammed Elmusrati , “*Greenhouse Monitoring with Wireless Sensor Network*”.University of Vaasa