

# Data retrieval from SCADA – By Somnath Bera

- The 16 MW solar field generates in isolation...
- The live generation data is not available.
- The Irradiation level is not known.
- The maintenance strategy is ineffective.
- Things were going by default.
- Loss was daily, weekly & yearly.

## In memoriam



The author dedicates this article to his mother, Sefali Bera, whose death deeply affected him and the community where she lived in Howrah, India. She was a great source of inspiration for his articles.

*I dedicate this session to my mom who was an avid follower of my articles & blogs that appeared all over the world*

# Data retrieval from SCADA – By Somnath Bera

- This powercos had a very old 16 MW solar generating field.
- The generation was by default [no management intervention as no data]
- The cleaning schedule, the angle adjustment schedule were all ineffective
- The efficiency loss was daily , weekly & yearly. Many strings were out of service.
- The demand for live data was huge.
- Demand for – present generation, present irradiation & days generation.
- The communication medium was only cellular phone.
- The arial distance was only 4550 meter ,vide one straight highway and two mining colonies.

# Remote telemetry – cheap yet efficient!



SCADA system & Windows XP Operating System

We thought that once we can get the data we can pull it up.

The SCADA was old and without any warranty.

No spare SCADA card to pull data out of the system [hydel like]

Even we tried to install *Capture2Text\_CLI.exe* or *Capture2Text.exe* an windows program to select data from a part of the SCADA screen but the warranty issue stopped us.

*Only option left was – Image analytic*

*SCADA - Supervisory Control and Data Acquisition*

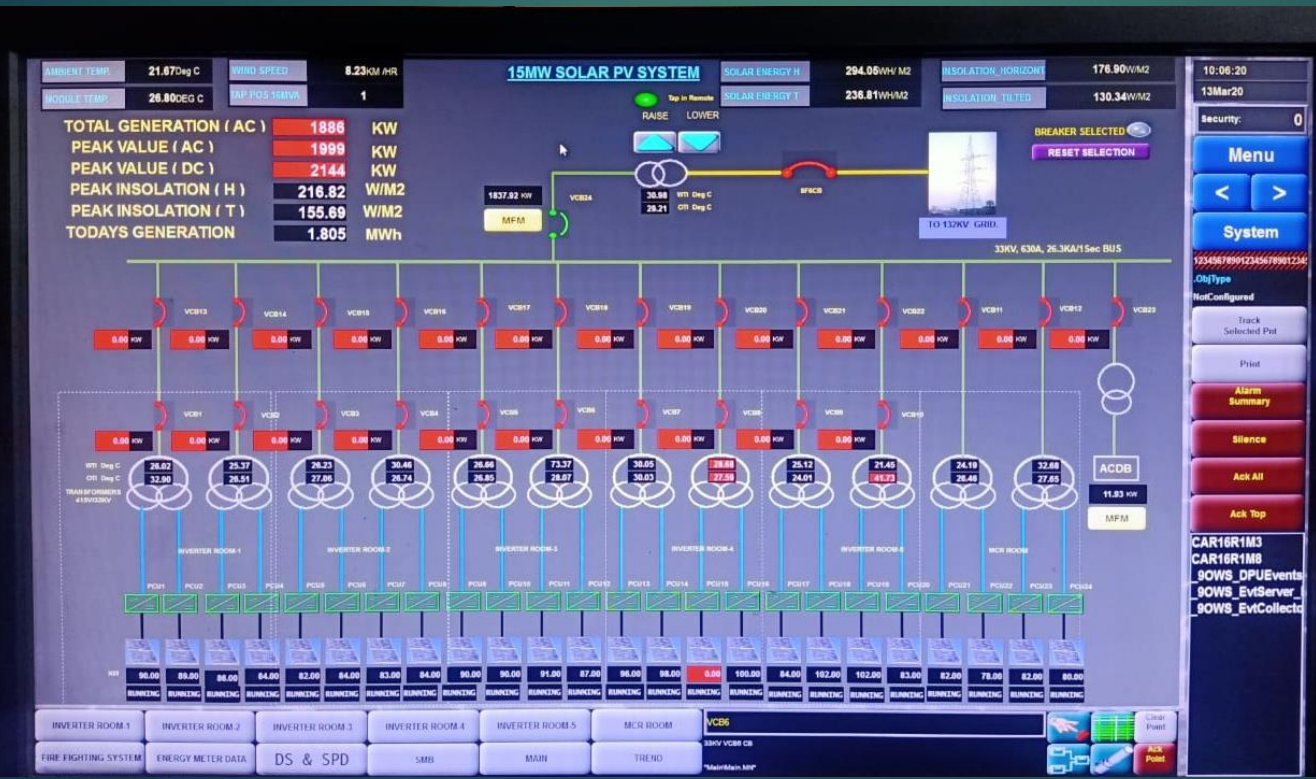
# SCADA – Image capturing.

To capture a moderate quality Image of the SCADA Screen

Analyse the image to capture 3 live data – Instant Generation, Instant Irradiation & cumulative MWH generation.

Python on a Desktop with an USB webcam & USB LoRa to work...

EAST text database module for speedy image detection & PySerial for uploading the data.



# Software required – all opensource software

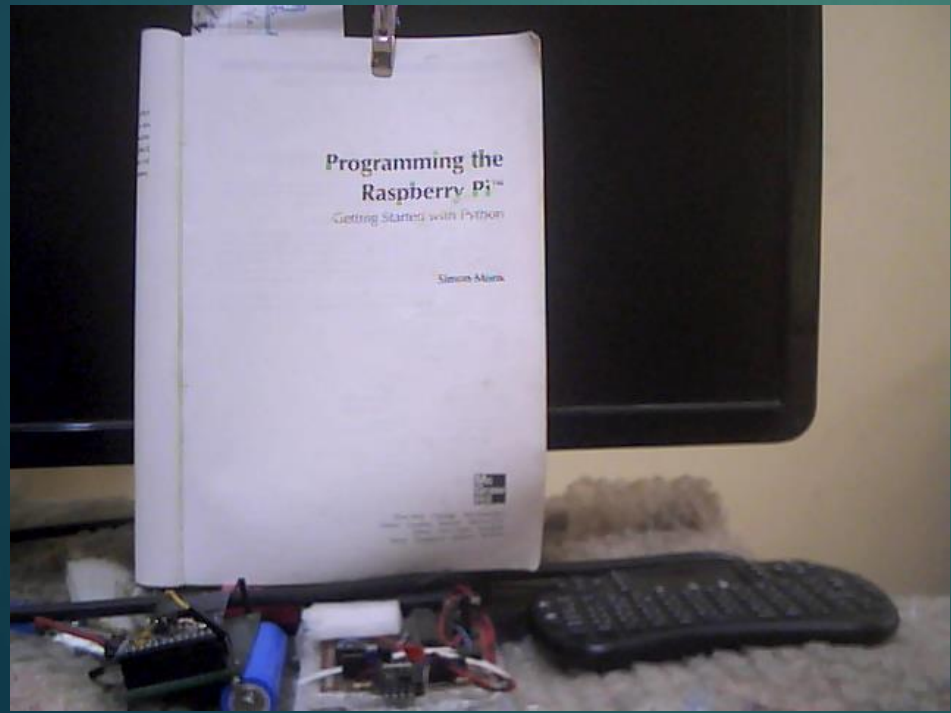
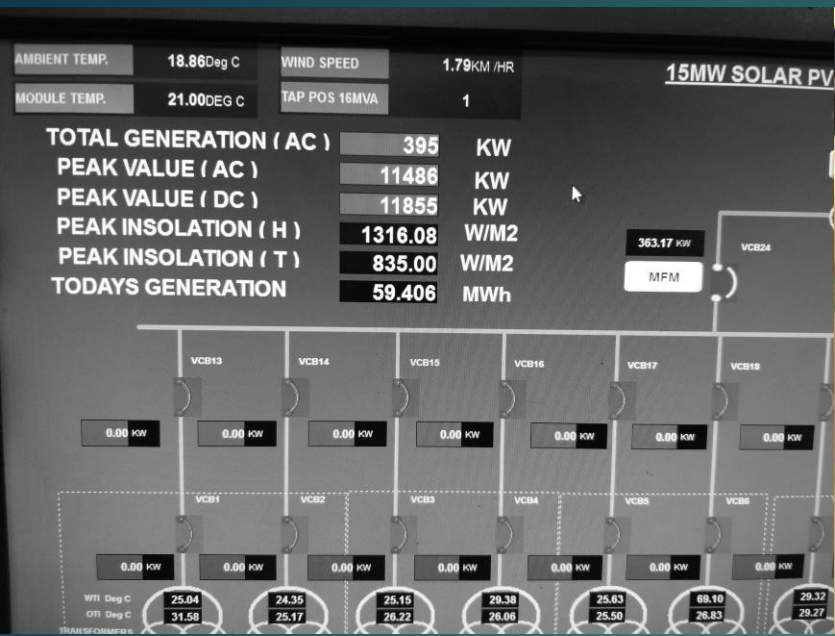
```
$ sudo apt-get update
$ sudo apt-get install imagemagick -y
$ sudo apt-get install python
$ sudo apt-get install python-minimal
$ sudo pip install pillow
$ sudo pip install numpy
$ sudo pip install pytesseract
$ sudo apt-get install tesseract-ocr
$ sudo apt-get install opencv-python
```

OpenCV's EAST (Efficient and Accurate Scene Text Detection - [frozen\\_east\\_text\\_detection.pb\\_](#)) is to be downloaded from Internet.

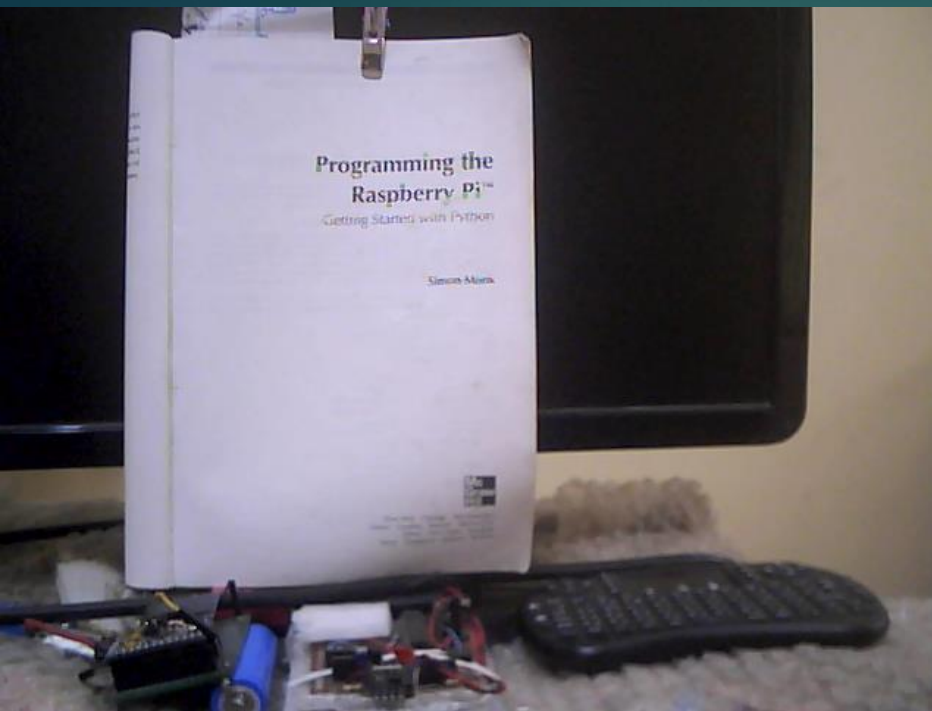
In Ubuntu 18 or above you may face that 'snap' is not allowing to install.

```
$ sudo snap changes
$ sudo snap abort xyz [snap process ID]
```

# Deployment of the process



# Deployment of the process



```
GtkTerm - /dev/ttyUSB1 9600-8-N-1
File Edit Log Configuration Control signals View Help
2:Programming 1
3:ing the
1:Raspberry Pi
2:Programming
3:ning the
1:taspberry PF
2:Programming
3:ing the
1:Raspberry P;
2:Programming
3:ing the
1:Raspberry Pi
2:Programming |
3:ing the
1:Raspberry P:
2:Programming
3:ing the
1:Raspberry Pi
2:Programming |
1:Raspberry Pi
2:Programming t
1:Raspberry Pi
2:Programming t
1:Raspberry Pi
2:Programming
1:Raspberry PB?"
2:Programming ti
1:Raspberry Pi
2:Programming ti
1:Raspberry Pi
2:Programming
3:ling the
```



## BOM:

02 nos UART 858\* MHz transceivers – USD \$8.25\*2 = USD \$16.5  
[\* Varies from country to country ]

One computer with python enabled and additional opensource softwares.

# Broad Schematic

- Image Capture by Computer Vision [CV2] →
  - Convert to Grey scale ->
  - Pass on to the Frozen NN ->
  - Apply EAST - Efficient and Accurate Scene Text Detection NN ->
  - Apply Sigmoid probability ->
  - Reconfigure to roi [image] ->
  - Read the text by pytesseract.image\_to\_string ->
  - Transmit the text to the remote receiver using LoRa ->
- Go to Image Capture ->

```
Pytesseract.image_to_string(image)
```



**EFY link...**

<https://www.electronicsforu.com/electronics-projects/prototypes/scada-too-old-too-scary>

# Post deployment issues...



**Actual deployment:** Image analytic is a heavy duty work. Even without putting any delay statement in anywhere it process image once in every 10 seconds in a Core I-3 grade old desktop. To make it once per 2 seconds you have to deploy Core i5 grade PC.

During actual deployment we found that for cheap quality camera, the lighting creates an issue. What becomes clear during day hours becomes unclear during night time. However, with good quality camera the problem was overcome.

Nonetheless. Our requirement was during day hours only [solar PV generation] therefore, with a cheap camera our work continued till a good quality camera was installed by IT department. First we ran it from a desktop PC later we planned to move to a Raspberry Pi computer. The software remains the same.

With direct data & live flow the management was careful about losses and remedial measures & prompt actions.

Success of this project is a testimony that using image processing & text detection, data can be gathered from many unthinkable propositions...



Questions ?