Raspberry Pi based EPICS Embedded Control System for Deflector Conditioning System of Superconducting Cyclotron

- Siddharth Vardhan Pratihast
Variable Energy Cyclotron Centre,
Department of Atomic Energy,
India

Why We Need Deflector Conditioning?

Super-Conducting Cyclotron (K500) has two Electrostatic Deflectors Installed to Extract the Accelerated Beam from the Outer Radius.

Voltage Holding Capability of the Electrostatic Deflectors had Been One of the Limiting Factors of Extracting Higher Energy Beams

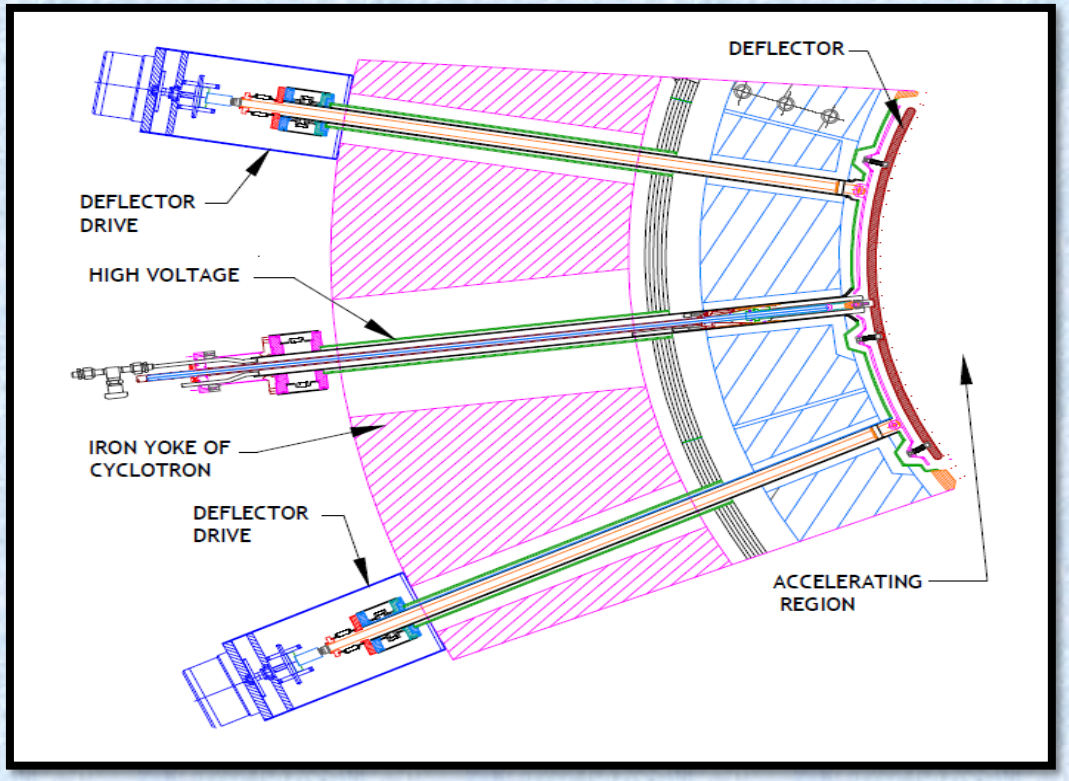
Voltage Holding Capability can be increased by conditioning the Deflector separately outside the cyclotron

Distance between Septum and Deflector Plate is 6 mm

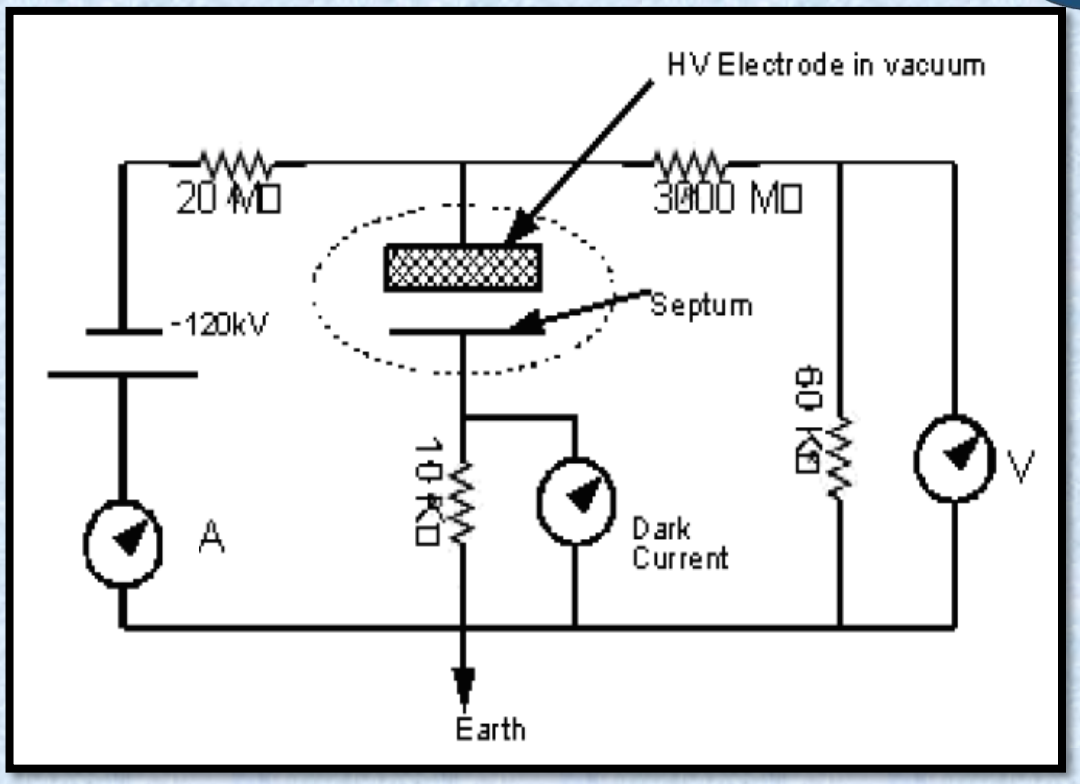
To Accelerate a Neon Beam of 50 MeV, an electric field of 81 KV/cm is required between Deflector and Septum which can be achieved by a Voltage of around 60 KV

It has been observed that if we Condition the Deflector to 80 KV, it becomes Operational for around 60-65 KV

Deflector Conditioning Setup

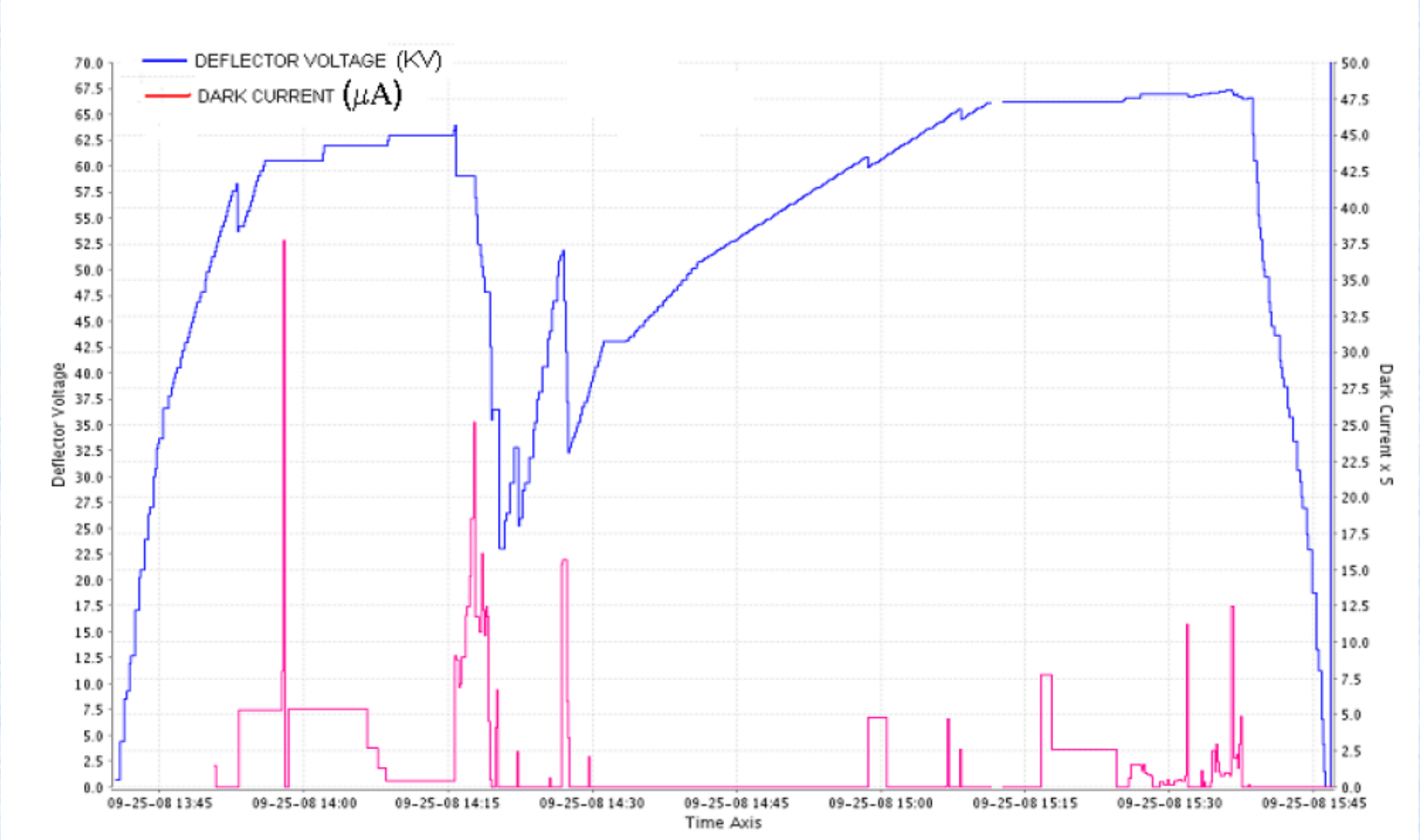


SCHEMATIC OF A WORKING DEFLECTOR

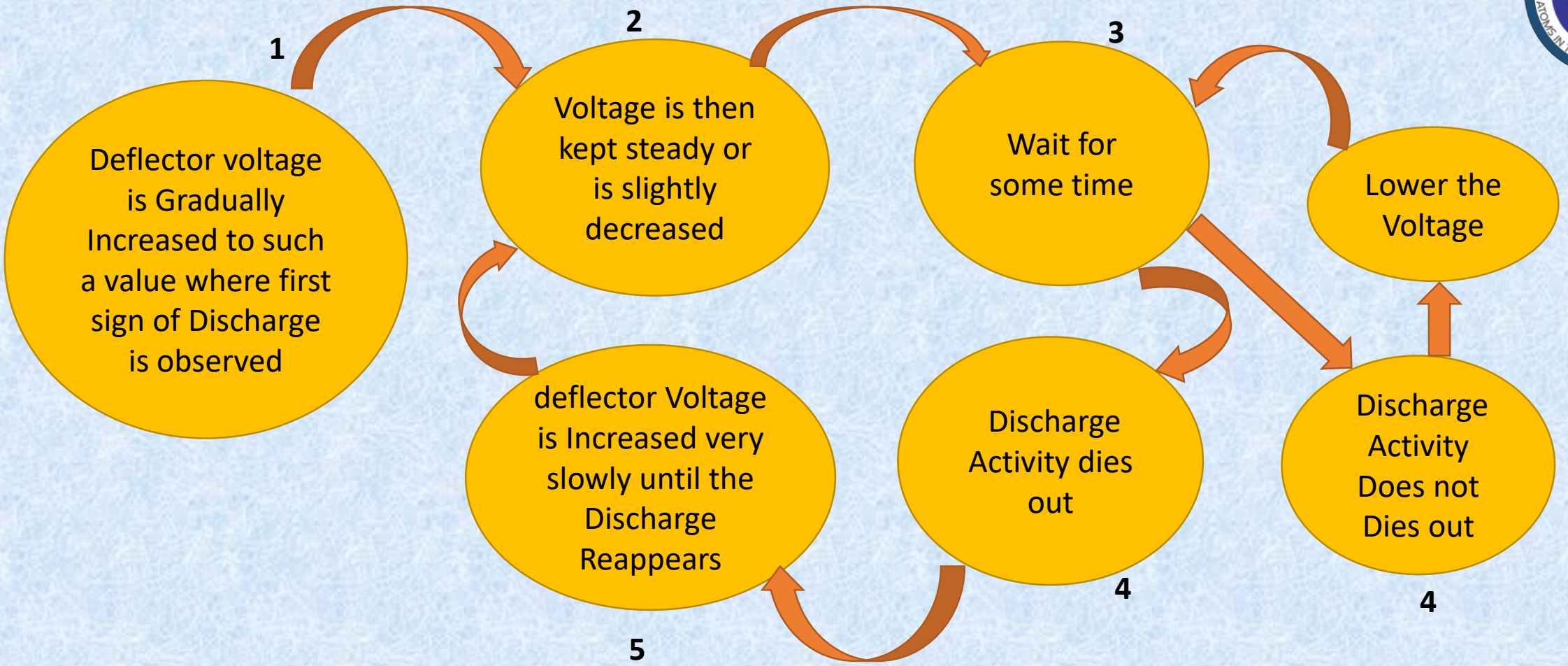


CIRCUIT DIAGRAM FOR CONDITIONING

Deflector Voltage and Current with respect to time Axis

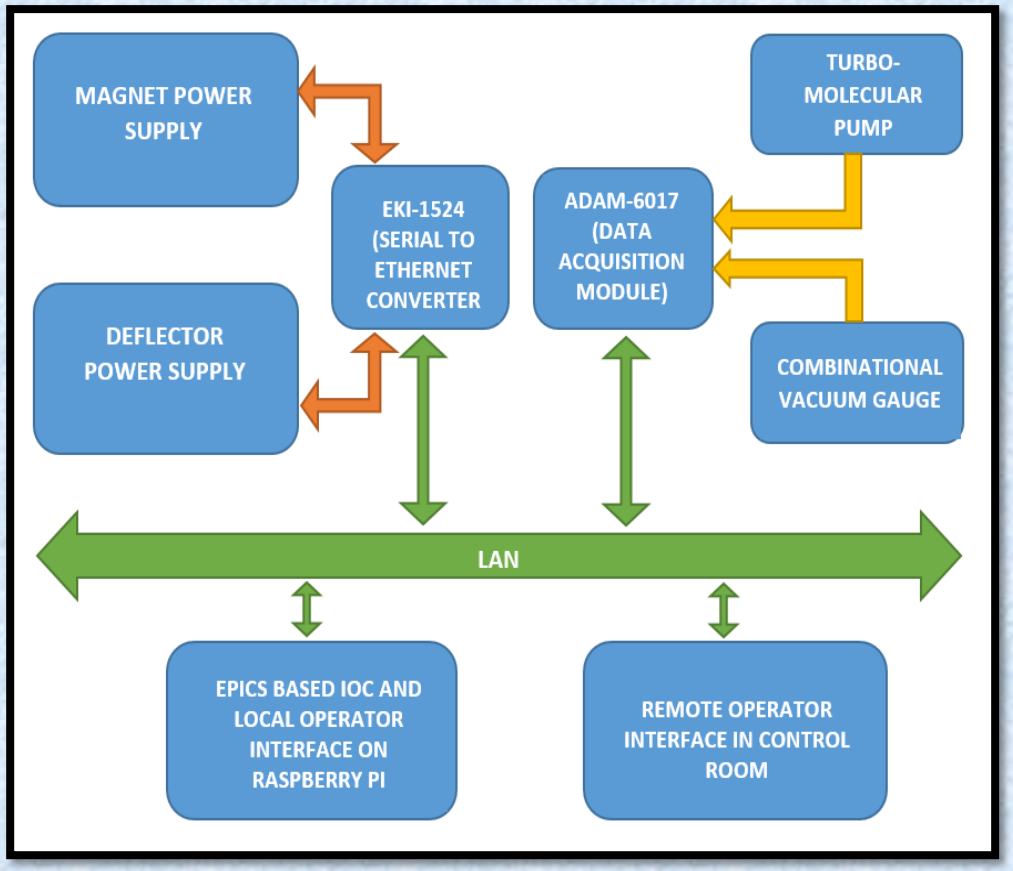


Steps for Conditioning

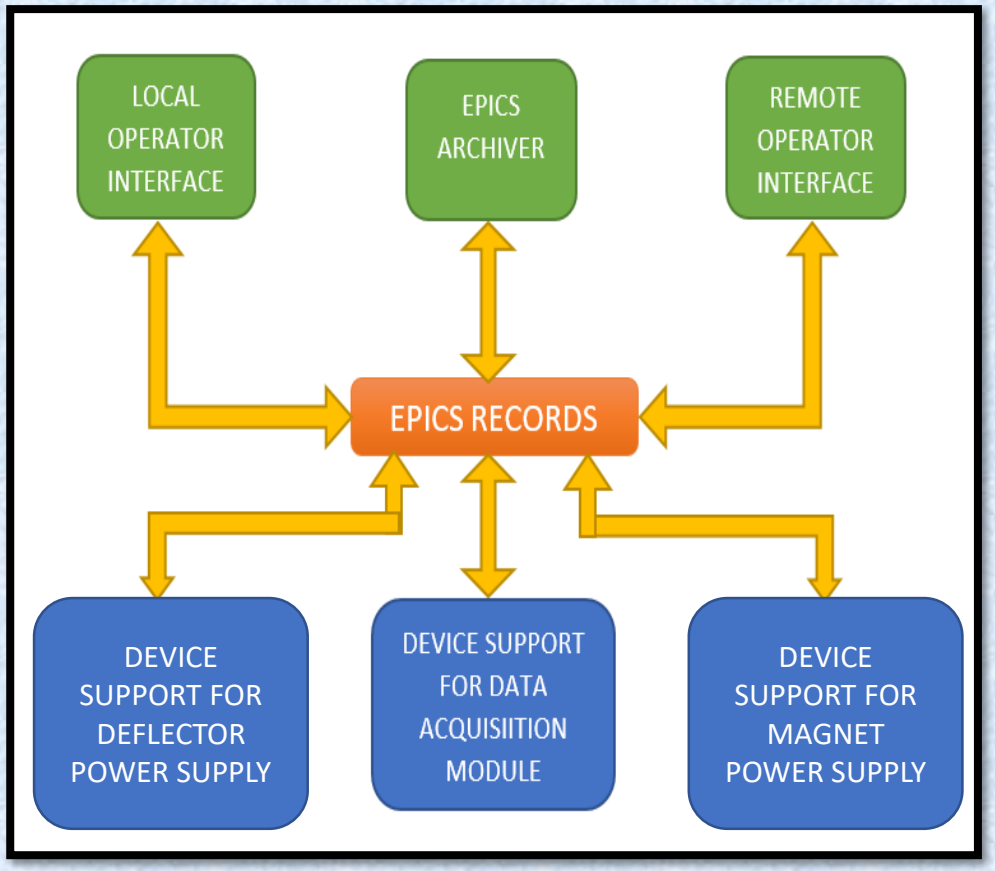


This Process is Repeated until Discharge Does not Die out Further. After which it can be said that Deflector is conditioned to that particular Voltage.

Conditioning Test Stand Setup Schematic



System Interconnection



Implementation of EPICS Based Architecture

Why Raspberry PI?

1.4GHz 64-bit ARM based quad-core processor, dual-band wireless LAN

4 GB RAM

Low Cost General Purpose Computer

Since Open-Source, Lot of online support available

Interactive Linux based Raspbian OS

Does not Require Cross Compilation. EPICS Application can be build directly over RaspberryPI

Occupies Very Less Space

Low Power Consumption

Low maintenance required

EPICS IOC kept in start-up script can be restarted by just rebooting the device easily

Note: RaspberryPI should not be plugged out of power while running. This action can corrupt the SD card.

Running an EPICS IOC over Raspberry PI

Phoebus Based GUI and EPICS IOC Running over RaspberryPI

- **Raspbian OS was loaded through SD Card**
- **Necessary Proxy Settings for Internet Access**
- **Setting Various EPICS and NON-EPICS Environment Variables**
- **An EPICS Base of version 3.15.9 was build over RaspberryPI**
- **An EPICS Application was build and tested**
- **JAVA-FX was installed which was a dependency of Phoebus**
- **Phoebus (A Channel Access Client) was installed for ARM architecture.**
- **A Phoebus Based GUI Developed for Local Monitoring on RaspberryPI Itself**

Why EPICS?

EPICS or Experimental Physics and Industrial Control System Framework uses Channel Access Protocol

Network-based client-server model where the data element is a process variable

Can Be Accessed from any channel access client connected in network

Entire set of Process variables establish a Distributed Real-time Database

Requires Access to Process Variables

Provides Access to Process Variables

A RELIABLE AND ECONOMICAL SOLUTION FOR MANAGING MANY SYSTEMS FROM A SINGLE SCREEN !

An Open-Source Control System Architecture with a good Community Support

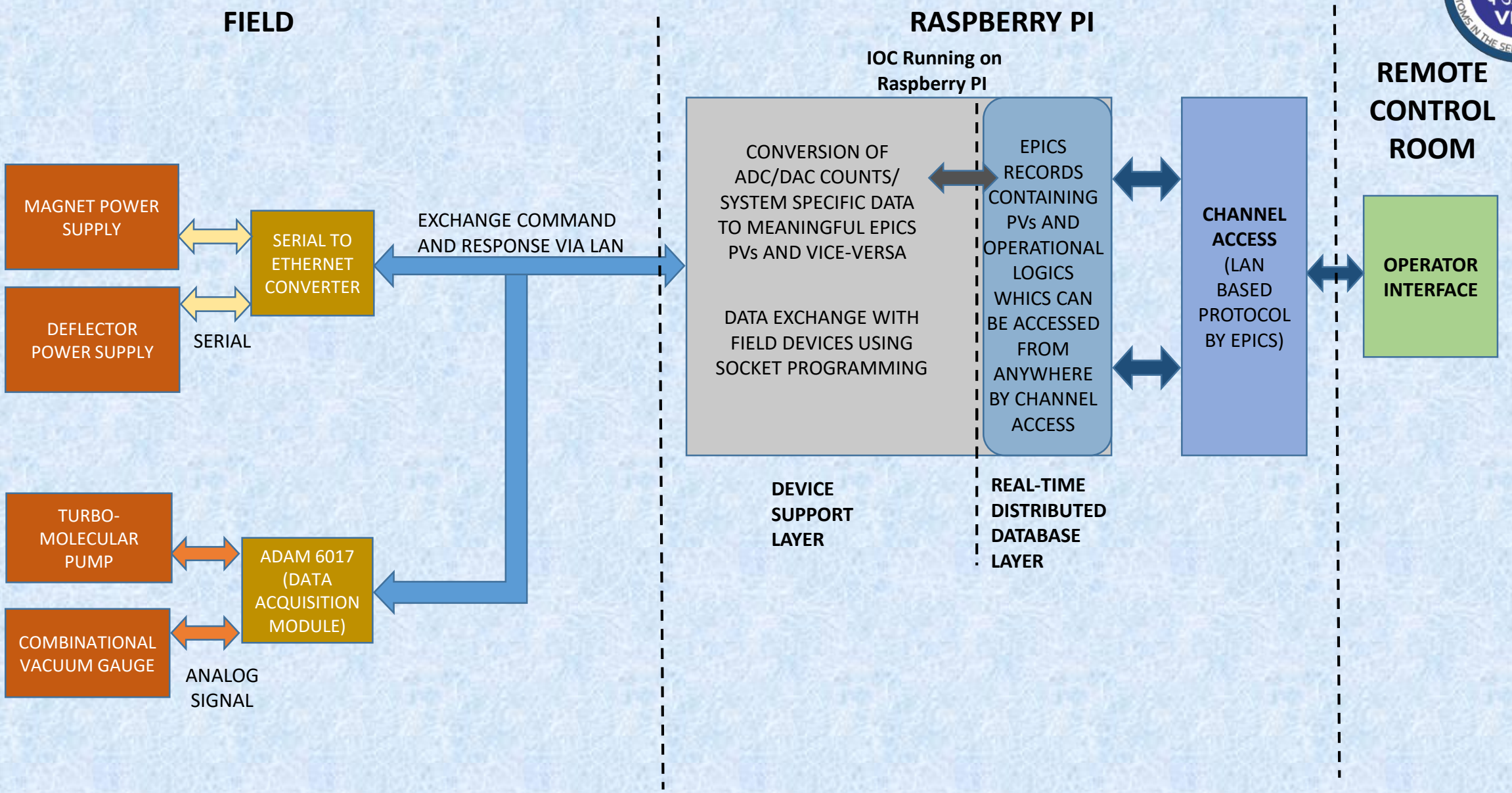
Three Layer Architecture consisting of Device Support, Database layer and the Operator Interface

Database Layer Contains many Records. Each records has various fields. Process Variable = Record Name + Field Name

Modular Control System Design (We can add the Individual IOCs from multiple locations in the Network to build a Larger one!

Drag and Drop Based GUI Building Applications such as CS Studio and Phoebus !

Implementation of EPICS Based Architecture



EPICS Working Explained

1. When **Operator** Sets the Voltage in **OPI**, it gets loaded into an **EPICS** Record through **channel access**

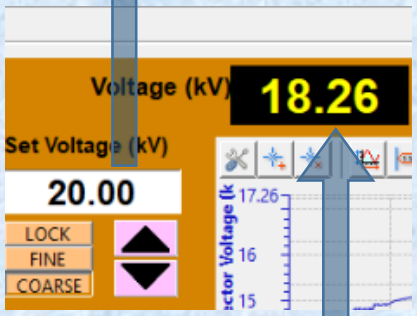
```
record(ai,"DEF:VOLT:OPI")
{
  field(SCAN, "Passive")
}
```

2. Operational Logic and **Soft Interlock** Implemented in **Intermediate Records** checks and processes the data

```
record(calc,"DEF:...")
{
.....
}
.....
```

```
record(ao,"DEF:VOLT:SET")
{
  field(SCAN, "Passive")
  field(DTYP, "deflectorPS")
  field(OUT, "#L1 @1")
}
```

```
record(ai,"DEF:VOLT:READ")
{
  field(SCAN, "Passive")
  field(DTYP, "deflectorPS")
  field(INP, "#L1 @2")
}
```



3. After Processing the Intermediate Records, **OUTPUT Record** passes the value to the **device support**

4. In the **Device support** file, first function forms a proper **device specific command** to send the **new desired voltage**

```
//ASYN Based Device Support
Set_Volt_function1( arguments)
{
.....
}
Read_Volt_function2( arguments)
{
.....
}
```

```
#Socket Communication managed by ASYN
IPPortConfigure
("L1","192.168.0.87:5300")
```

5. The **Command** is sent through the **Socket (IP Address +Port)** with which **device** is connected.

6. Serial to Ethernet Converter does the required conversion and **Power Supply** Updates the **new Voltage Value**

IP ADDRESS:192.168.0.87

DEFLECTOR POWER SUPPLY

7. Raw Voltage Reading Response is received through same path and suitable conversion is performed by **second function** in **device support**

8. Device Support Loads the input voltage reading to an INP**UT Record** which is accessed through the **OPI** via **Channel Access**

Key Features of OPI Developed and Interlocks

Operational Logic and Interlock is Implemented in EPICS Record Level

Deflector Power Supply with Ramp Control

Ramp Menu Provided for Operator

Continuous Monitoring of **Vacuum Scanned in Every Second and related Interlock**

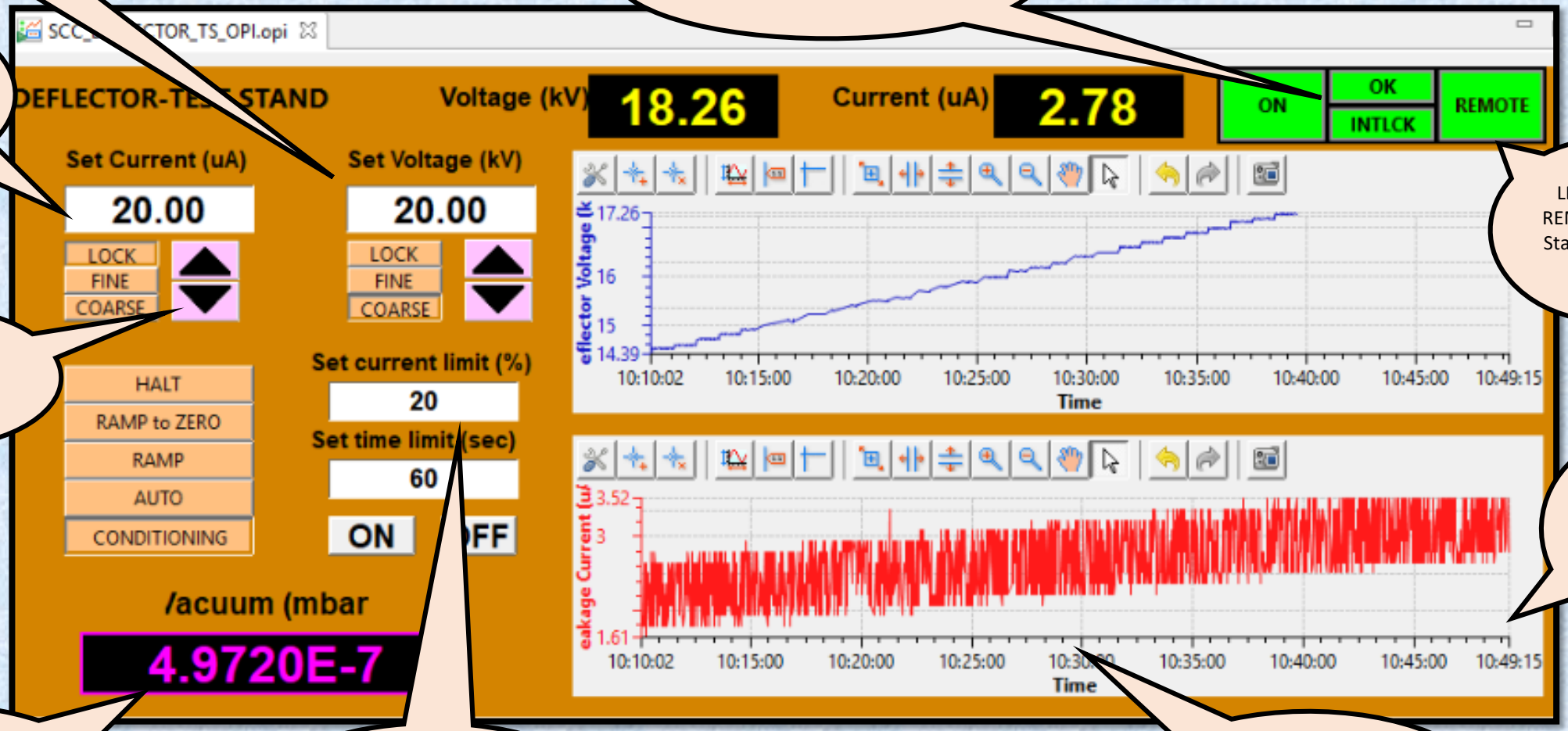
Door Interlock for Human Protection from High Voltage

LED showing REMOTE/LOCAL Status of Power Supply

Deflector Voltage Plot With respect to time

Leakage Current Limit Set provided for Operator

Leakage Current monitoring with Respect to time



Results we have Achieved

Magnetic Field of 700-1000 Gauss was applied

Time Taken for Conditioning: 10-15 Days

Deflector Voltage : 80 KV

Deflector is been conditioned since 3 months without any down-time by existing system

Thank You For Your Attention

- Siddharth Vardhan Pratihast
Variable Energy Cyclotron Centre,
Department of Atomic Energy
India
sv.pratihast@vecc.gov.in