Palm Vein Device

KRC20664

GBHU0003535

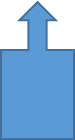
GBHU000064

**UNIT1** **UNIT2**

Weigh Bridge

CAM 2

CAM1

Direction of Train Movement

Wagon camera 3

**PRINCIPLE OF OPERATION.**

**UNIT 1** == OLD TYPE CONTAINERS == GBHU000064

**UNIT 2** == NEW TYPE CONTAINER == GBHU0003535

1. The Train is moving from left to the right through the loading point /weigh bridge
2. As the train enters the weigh bridge, Both Unit 2 and UNIT 1 will pass past camera 1 , meaning that there are chances both containers might be read at Camera 1.
3. Unit 2 is the only unit that will get to camera 2 point meaning it shall also be read by camera 2.
4. The same applies we can also have 2 more cameras to make it 2 cameras per unit (The more accuracy obtained ) Installed at?
5. Let’s stick to we just have the 2 unit cameras and the 1 wagon camera
6. Wagon number camera is just one but the same case it can be two cameras as well – Though not necessary.

**SCENARIO CASE 1**

The container capture engine is always armed and captures containers in real time as they pass by the camera view.

In our case, let us assume at CAMERA 1 the below readings happened as train entered loading point.

**UNIT 1 READING** = GBHU000064 (**Full container number read**)

**UNIT 2 READING** = GBHU0003535 (**Full Container number read**)

The train proceeds and positions by the STX ,at this particular point UNIT 2 Is adjacent Camera 2 and UNIT 1 is adjacent Camera 1.

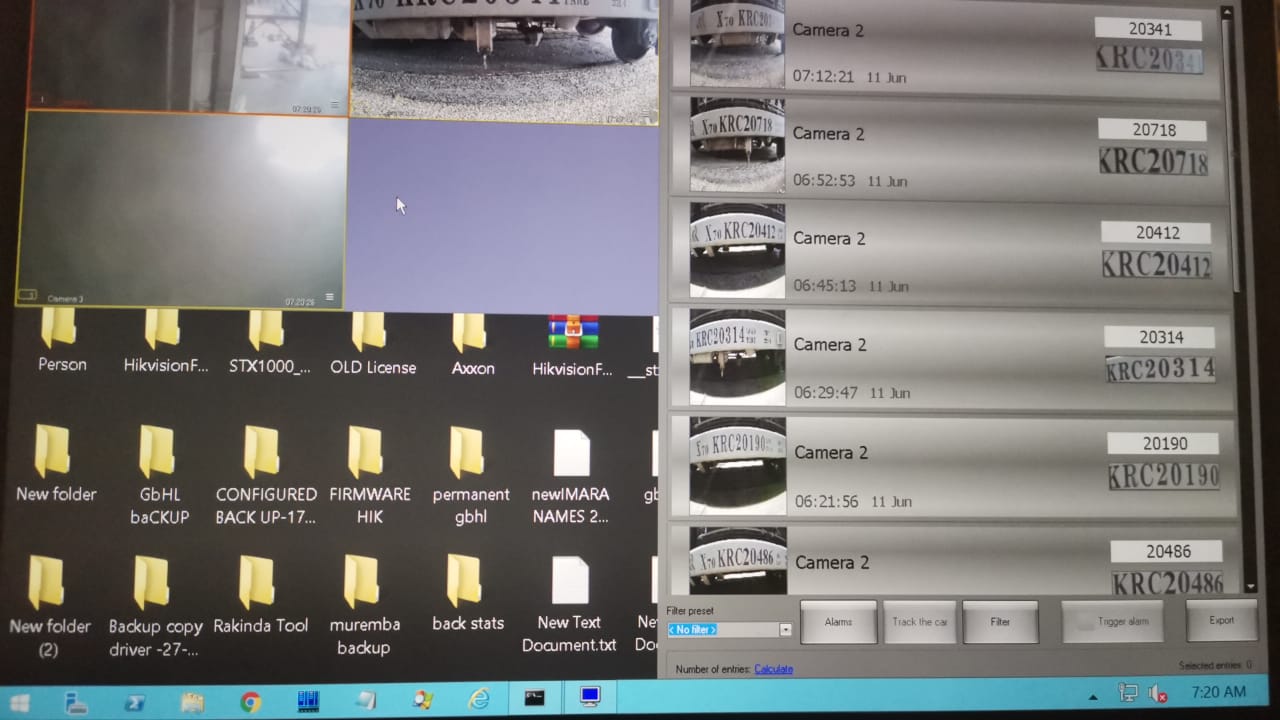
For example basis let us at this particular point when train is already positioned the below readings are read by cameras.

**CAMERA 1** = GBHU00006 (**less a digit**)

**CAMERA 2** = GBHU000353 (**Less a digit**)

And let’s say more unit variants were captured by container cameras. The logic shall be below

1. Axxon shall iterate and loop through all captured results from both camera 1 and camera 2 to ensure that we remain with two Unit unique numbers ( **Logic used here is the unit number with more digits is the correct one for the unique set**)
2. After above automated sort by Axxon completes, Axxon will end up with two unit numbers.
3. Axxon will take the two units numbers and scan them against the Unit master copy to verify if it exists in master copy. (Sharing masters with AXXON?)
4. If exists in master copy Axxon will have this two units in script memory for next step.
5. Ones GBHL user presents his Palm , Axxon will get the user ID
6. Axxon will then send above three data to logstar Plus wagon number by Api call (GBHU000064 , GBHU0003535, KRC20664,D0012)
7. Axxon will check STX Not blocked and if correct call Api and expect a success or error
8. IF success Axxon reads weight 1
9. Container units above are retained in memory but user ID is dropped
10. After train loads, the user scans palm to obtain his ID
11. Check number 7 occurs and API is called after which we expect error or success
12. If success we read weight number 2
13. After reading weight number 2 , Script clears all data in memory and expects the next set of container and wagon recognitions (if by some reason we want to change something we start all over again? Can next successful API call clear date? )
14. Above cycle repeats until entire train is loaded

****

**DISTINGUISHING BETWEEN OPEN TOP AND NORMAL UNITS**

The open top Type is one entire unit, the unit has three code writings that are located in below locations:-

1. Top Right (Not at camera view)
2. Top left (Not at camera view)
3. Centre (**Within Camera view**)



The Wagon number on the Open Top is in the same view of the same camera doing wagon recognition for other normal wagons.

The open top Wagon number has a unique identifier in that the wagon number consist of 1XXXX pattern , meaning that Open Top wagon numbers all start with number 1

The Open Top wagon unit numbers also have smaller characters compared to the normal wagons. The logic will be built on this elements as follows

1. If recognition received from camera this has pattern 1xxxx and less character type it means the unit loading is Open top type
2. Axxon Captures the respective wagon number **e.g. 10255**
3. Axxon builds in memory two unit numbers equivalent to Wagon number
4. Gbhl operator presents palm , Axxon captures user ID
5. Axxon populates above 4 data points in script (10255 ,10255,10255 ,D0012 ) Corresponding to unit 1, unit 2 , wagon number and driver code (Restrictions on having the same 2 unit numbers we have to find solution to this either we put one as KRC10255\*) This would mean having both on db or we remove the validations of not having two similar units)
6. Axxon checks Stx sensors give no obstruction meaning train on weigh bridge
7. Axxon calls logstar Api with above dataset
8. Logstar verifies and returns success or error
9. If Success , Axxon reads Weigh 1 and discards the user ID but retains the other dataset in memory
10. The open top unit loads with cargo
11. Gbhl user scans his Palm
12. Axxon calls Api second time
13. Logstar verifies received data
14. Logstar returns success or error
15. If success , Axxon reads second weight and clears all date in memory
16. Axxon awaits next Open top or GBHU unit loading
17. The cycle continues with either logic of OT or NORMAL as decided by the logic