SICK Inspector P611 - FTP Image Recording Setup Guide

# ✅ Objective:

Enable SICK Inspector P611 camera to save inspection result images (PASS/FAIL) to a Raspberry Pi using FTP, and use the images for logging or reports.

# 🔧 Tools Used

- FTP Protocol: Used to transfer image files from the camera to Raspberry Pi

- vsftpd: Lightweight FTP server installed on Raspberry Pi

- Raspberry Pi OS: Acts as the FTP server for storing images

- SICK Inspector P611: Industrial smart camera with built-in job handling and image recording features

# 🔧 Part 1: Set Up FTP Server on Raspberry Pi

1. Install FTP Server

sudo apt update  
sudo apt install vsftpd -y

2. Edit FTP Configuration

sudo nano /etc/vsftpd.conf

Ensure the following lines are set:

write\_enable=YES  
local\_enable=YES  
chroot\_local\_user=YES  
allow\_writeable\_chroot=YES

Save and exit with Ctrl+O, Enter and Ctrl+X

3. Restart FTP service

sudo systemctl restart vsftpd

4. Create Directory for Images

mkdir -p /home/pi/asmscamera/images  
chmod 777 /home/pi/asmscamera/images

# 🔢 Part 2: Configure SICK Inspector P611

Login to the camera's web interface using its IP (e.g., http://192.168.0.1).

Navigate to Jobs → Communication → Image Collection and set the following:

|  |  |
| --- | --- |
| Field | Value |
| Collection mode | FTP |
| File type | PNG or JPG |
| FTP server IP address | 192.168.0.5 (Pi IP) |
| FTP server port | 21 |
| FTP username | pi |
| FTP password | your Pi password |
| Output folder | asmscamera/images/ |
| Sort into job folder | ✔ Enabled |
| Append string result | Result String/Output String |
| Save condition | Always / On Fail |

⚠️ Do not use special characters (like `\_\_`) in folder names.  
⚠️ Do not start folder names with `/`.

# 📊 Part 3: Test FTP Connection

From Raspberry Pi:

ftp 192.168.0.5

Login:  
Name: pi  
Password: [your password]

Test file transfer:

cd asmscamera/images  
put test.txt  
bye

# 🚀 Part 4: Test Camera Recording

Run a job manually or via Python socket script.  
After job is triggered and image recorded:

ls ~/asmscamera/images

# 📄 Optional: Logging & Report

You can modify your Python script to log:  
- Job number  
- Time of trigger  
- Pass/Fail result  
- Image file name or path

# ❌ Troubleshooting

❌ Permission denied: Check folder permissions: chmod 777 -R ~/asmscamera/images

❌ Disallowed characters: Folder names should be alphanumeric only

❌ No image saved: Ensure Save Condition is set to Always or On Fail

❌ Output folder incorrect: Do not start with / in Output folder

# 🚀 Result:

Once correctly configured, the camera will automatically store pass/fail images to the Raspberry Pi for inspection reporting.  
  
How FTP Works — Simple Steps

### 🎯 Purpose:

To move files **from one system to another**, typically:

* From a **client** (e.g., a camera or PC)
* To a **server** (e.g., a Raspberry Pi or Linux system)

## ⚙️ Components:

| Component | Role | Example |
| --- | --- | --- |
| **FTP Client** | Sends/receives files | SICK Inspector P611 camera |
| **FTP Server** | Stores or provides access to files | Raspberry Pi with vsftpd |

## 📡 FTP Communication Steps

### 1. ****Client initiates connection****

* The client (camera) connects to the server (Pi) on port **21**
* Uses **TCP/IP**

### 2. ****Login****

* The client sends a **username & password**
* The server checks permissions

### 3. ****Navigation****

* The client sends commands:
  + cd folder/
  + put filename (upload)
  + get filename (download)
  + ls (list directory)

### 4. ****Data Transfer****

* FTP opens a **data connection**
* Files are transferred as:
  + **Binary** for images, programs, videos
  + **ASCII** for text files

### 5. ****Disconnect****

* The client closes the session with a QUIT command

## 📦 Example in Your Case:

| Action | Who | What Happens |
| --- | --- | --- |
| Connect | Camera | Logs in to 192.168.0.5:21 |
| Authenticate | Pi | Accepts user pi, password 123 |
| Store | Camera | Uploads .png image |
| Save | Pi | Stores file in /home/pi/asms\_camera/ftpimages/ |
| Done | Camera | Ends session until next job |