Okay, here is the step-by-step guide translated into English for running the project in Visual Studio Code from scratch.

**Setup Guide for VS Code: Pill Project Pilot GUI**

This guide assumes you have the following Python files: main.py, variables.py, calib.py, robot\_comm.py, ultimate.py, and test.py. The main logic uses ultimate.py for detection, while test.py appears to be an older or alternative version and is not directly run by main.py.

**Step 1: Install Prerequisites**

1. **Python:** Ensure you have Python 3.6 or newer installed. Download from [python.org](https://www.google.com/url?sa=E&q=https%3A%2F%2Fpython.org%2F). During installation, **check the box "Add Python to PATH"**.
2. **Visual Studio Code:** Download and install VS Code from [code.visualstudio.com](https://www.google.com/url?sa=E&q=https%3A%2F%2Fcode.visualstudio.com%2F).
3. **Python Extension for VS Code:** Open VS Code, go to the Extensions view (icon on the left sidebar or Ctrl+Shift+X), search for "Python" (by Microsoft), and install it.

**Step 2: Prepare the Project**

1. **Create a Folder:** Create a folder on your computer for the project (e.g., PillProjectGUI).
2. **Copy Files:** Place all six .py files (main.py, variables.py, calib.py, robot\_comm.py, ultimate.py, test.py) into the created PillProjectGUI folder.

**Step 3: Open Project in VS Code**

1. Launch Visual Studio Code.
2. Go to File > Open Folder....
3. Select the PillProjectGUI folder you created and click "Select Folder".
4. VS Code will open the folder, and you should see the project files in the "Explorer" panel on the left.

**Step 4: Set Up Environment and Install Dependencies**

1. **Open Terminal:** In VS Code, open the integrated terminal: Terminal > New Terminal.
2. **(Recommended) Create a Virtual Environment:** This isolates project dependencies. In the terminal, run:
3. python -m venv venv

1. **Activate the Virtual Environment:**
   * **Windows (cmd/PowerShell):**
   * .\venv\Scripts\activate

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

IGNORE\_WHEN\_COPYING\_END

* + **macOS/Linux (bash/zsh):**
  + source venv/bin/activate

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

IGNORE\_WHEN\_COPYING\_END

* + You should see (venv) appear at the beginning of your terminal prompt after activation.

1. **Install Libraries:** Run the following command in the **activated** terminal (this installs all necessary external libraries mentioned in the files):
2. pip install opencv-python numpy pyModbusTCP PyQt5 pandas scipy

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

IGNORE\_WHEN\_COPYING\_END

**Step 5: Configure Paths (CRITICAL STEP!)**

1. **Open variables.py** in VS Code.
2. **Locate the lines:** CALIBRATION\_VIDEO\_PATH and WORKING\_VIDEO\_PATH.
3. **Change Paths:** Replace the example paths (e.g., r'C:\Users\Stas\...') with the **actual paths** to your video files for calibration (cali.mp4) and the working mode (e.g., circular\_white.mp4) **on your computer**.
   * Use raw strings (r'...' as in the example), double backslashes (\\), or forward slashes (/) for paths on Windows to avoid issues.
   * Example: CALIBRATION\_VIDEO\_PATH = r'D:\MyVideos\Pills\cali.mp4'
4. **Select Working File:** Comment out (add # at the start of the line) the unused WORKING\_VIDEO\_PATH lines and uncomment the one you want to use by default.
5. **(Optional) Source:** Check the source\_type variable ("video" or "camera") and CAMERA\_INDEX (if using a camera).
6. **(Optional) Modbus:** If you plan to use robot communication, verify MODBUS\_TCP\_HOST and MODBUS\_TCP\_PORT.
7. **Save the file** (Ctrl+S).

**Step 6: Run Calibration (If Necessary)**

* Calibration is needed if you don't have the app\_data/camera\_calibration.npz file, or if you want to update it for your camera/conditions.
* The app\_data folder should be created automatically when the application starts if it doesn't exist.
* **Method 1 (Via GUI):** Skip this step, run the main application (Step 7), navigate to the "Calibration" tab, select the source ("Calibration video" or "Camera"), and click "Start Calibration".
* **Method 2 (Running calib.py directly - for testing):**
  1. Ensure CALIBRATION\_VIDEO\_PATH in variables.py points to your calibration video.
  2. Make sure your virtual environment (venv) is activated in the terminal.
  3. In the VS Code terminal, run:
  4. python calib.py

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

IGNORE\_WHEN\_COPYING\_END

* 1. Follow any on-screen instructions or wait for the process to complete. Results will be saved to app\_data/camera\_calibration.npz.

**Step 7: Run the Main Application**

1. Ensure your virtual environment (venv) is active in the terminal. If not, reactivate it (see Step 4.3).
2. In the VS Code terminal, run the command:
3. python main.py

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487).Bash

IGNORE\_WHEN\_COPYING\_END

1. The "Pill Project Pilot GUI" window should appear.
2. Navigate to the "Working Mode" tab.
3. Select the source ("Working video" or "Camera"). If using video, double-check that the path selected in the GUI (or variables.py) is correct.
4. Click "Start". The video feed should appear, and object detection (if configured) should begin processing frames.

**Troubleshooting / Notes:**

* **ModuleNotFoundError**: You likely haven't installed all dependencies (repeat Step 4.4) or haven't activated the virtual environment (repeat Step 4.3) before running python main.py.
* **Video Not Opening / cap.isOpened() Error**: Double-check the video file paths in variables.py (Step 5) and in the GUI if you selected a different file. Ensure the files exist and are accessible. Try different path formats (r'', /, \\).
* **Camera Not Working**: Ensure source\_type = "camera" in variables.py and CAMERA\_INDEX is correct (usually 0 for built-in, 1+ for external). Check that no other application is using the camera and that Python/VS Code has permission to access the camera in your OS settings.
* **app\_data Folder Missing**: It should be created automatically. If not, check write permissions in the project folder.
* **Modbus Connection Failed**: Verify the IP address and Port in the GUI or variables.py are correct. Ensure the Modbus TCP server (robot) is running and accessible on the network from your computer. Check network settings and firewalls.
* **Performance Issues**: Video processing is resource-intensive. Parameters in variables.py (like TARGET\_WIDTH, SCALE) and the GUI sliders (Blur, Canny thresholds, etc.) affect performance.

Following these steps should get the project running in VS Code. The most common issues are incorrect paths and problems with the Python environment or dependencies.