.. _HomeAssistant-Driver:

Home Assistant Driver
====================

The Home Assistant driver enables VOLTTRON to read any data point from any Home Assistant controlled device.
Control (write access) is now supported for:

- Lights (state, brightness)
- Thermostats (state, temperature)
- **Smart Locks (NEW)**
- **Fans (state, percentage) (NEW)**
- **Motorized Curtains / Covers (state, position) (NEW)**

The following diagram shows interaction between platform driver agent and Home Assistant driver.

.. mermaid::

```
   sequenceDiagram
       HomeAssistant Driver->>HomeAssistant: Retrieve Entity Data (REST API)
       HomeAssistant-->>HomeAssistant Driver: Entity Data (Status Code: 200)
       HomeAssistant Driver->>PlatformDriverAgent: Publish Entity Data
       PlatformDriverAgent->>Controller Agent: Publish Entity Data

       Controller Agent->>HomeAssistant Driver: Instruct to Turn Off Light
       HomeAssistant Driver->>HomeAssistant: Send Turn Off Light Command (REST API)
       HomeAssistant-->>HomeAssistant Driver: Command Acknowledgement (Status Code: 200)
```

Pre-requisites

Before proceeding, find your Home Assistant IP address and long-lived access token from `here <https://developers.home-assistant.io/docs/auth_api/#long-lived-access-token>`_.

Clone the repository, start volttron, install the listener agent, and the platform driver agent.

- `Listener agent <https://volttron.readthedocs.io/en/main/introduction/platform-install.html#installing-and-running-agents>`_
- `Platform driver agent <https://volttron.readthedocs.io/en/main/agent-framework/core-service-agents/platform-driver

/platform-driver-agent.html#configuring-the-platform-driver>`_

Configuration

Each Home Assistant device requires:

1. A **device configuration file**
2. A **registry file**

Ensure that the ``registry_config`` parameter in the device configuration links to the correct registry file stored in VOLTTRON’s configuration store.

Device configuration
+++++++++++++++++++

.. code-block:: json

```
    {
        "driver_config": {
            "ip_address": "Your Home Assistant IP",
            "access_token": "Your Home Assistant Access Token",
            "port": "Your Port"
        },
        "driver_type": "home_assistant",
        "registry_config": "config://light.example.json",
        "interval": 30,
        "timezone": "UTC"
    }
```

Registry Configuration
++++++++++++++++++++++

A registry file can contain one or more Home Assistant entities.

Each entry includes:

- **Entity ID** (e.g., ``light.example``, ``fan.living_room``, ``cover.my_shade``)
- **Entity Point** – the state or attribute (e.g., ``state``, ``brightness``, ``percentage``, ``position``)
- **Volttron Point Name** – unique name inside VOLTTRON

- **Writable** – whether the point supports ``set_point``
- **Type** – int, float, boolean, string…

Attributes can be viewed in Home Assistant under:

Developer Tools → States

Example Light Registry

.. code-block:: json

```
    [
        {
            "Entity ID": "light.example",
            "Entity Point": "state",
            "Volttron Point Name": "light_state",
            "Units": "On / Off",
            "Units Details": "on/off",
            "Writable": true,
            "Starting Value": true,
            "Type": "boolean",
            "Notes": "lights hallway"
        },
        {
            "Entity ID": "light.example",
            "Entity Point": "brightness",
            "Volttron Point Name": "light_brightness",
            "Units": "int",
            "Units Details": "light level",
            "Writable": true,
            "Starting Value": 0,
            "Type": "int",
            "Notes": "brightness control, 0 - 255"
        }
    ]
```

Example Thermostat Registry
****

.. code-block:: json

```json
[
    {
        "Entity ID": "climate.my_thermostat",
        "Entity Point": "state",
        "Volttron Point Name": "thermostat_state",
        "Units": "Enumeration",
        "Units Details": "0: Off, 2: Heat, 3: Cool, 4: Auto",
        "Writable": true,
        "Starting Value": 1,
        "Type": "int"
    },
    {
        "Entity ID": "climate.my_thermostat",
        "Entity Point": "current_temperature",
        "Volttron Point Name": "volttron_current_temperature",
        "Units": "F",
        "Writable": true,
        "Type": "float"
    },
    {
        "Entity ID": "climate.my_thermostat",
        "Entity Point": "temperature",
        "Volttron Point Name": "set_temperature",
        "Units": "F",
        "Writable": true,
        "Type": "float"
    }
]
```

Example Smart Lock Registry (NEW)
****

.. code-block:: json

```json
[
    {
        "Entity ID": "lock.front_door",
        "Entity Point": "state",
        "Volttron Point Name": "front_door_lock_state",
        "Units": "0 = unlocked, 1 = locked",
        "Writable": true,
        "Type": "int",
        "Notes": "Front door smart lock"
        }
    ]
```

Example Fan Registry (NEW)
****

Fans in Home Assistant use ``fan.*`` entity IDs.

Supported:

- ``state`` → on/off
- ``percentage`` → 0 – 100 speed

.. code-block:: json

```
    [
        {
            "Entity ID": "fan.living_room",
            "Entity Point": "state",
            "Volttron Point Name": "living_room_fan_state",
            "Units": "On/Off",
            "Writable": true,
            "Type": "int"
        },
        {
            "Entity ID": "fan.living_room",
            "Entity Point": "percentage",
            "Volttron Point Name": "living_room_fan_percentage",
            "Units": "%",
            "Writable": true,
            "Type": "int"
        }
    ]
```

Example Motorized Curtain / Cover Registry (NEW)
****

Covers in Home Assistant include:

- Electric curtains
- Window blinds
- Roller shades

- Garage doors

Supported:

- ``state`` → open/closed (mapped to 1/0)
- ``position`` → 0 – 100%

.. code-block:: json

```
    [
        {
            "Entity ID": "cover.living_room_curtain",
            "Entity Point": "state",
            "Volttron Point Name": "curtain_state",
            "Units": "Open/Closed",
            "Writable": true,
            "Type": "int"
        },
        {
            "Entity ID": "cover.living_room_curtain",
            "Entity Point": "position",
            "Volttron Point Name": "curtain_position",
            "Units": "%",
            "Writable": true,
            "Type": "int"
        }
    ]
```

Transfer registry and configuration files into the VOLTTRON config store:

.. code-block:: bash

```
    vctl config store platform.driver light.example.json HomeAssistant_Driver/light.example.json
    vctl config store platform.driver devices/BUILDING/ROOM/light.example HomeAssistant_Driver/light.example.config
```

Running Tests
++++++++++++++++++++++

To run tests for the Home Assistant driver:

1. Create a toggle helper named ``volttrontest`` in Home Assistant:

Settings → Devices & Services → Helpers → Create Helper → Toggle

2. Run pytest from the VOLTTRON root:

.. code-block:: bash

```
    pytest volttron/services/core/PlatformDriverAgent/tests/test_home_assistant.py
```

If everything works, you will see all tests passed, including:

- Light tests
- Thermostat tests
- **Lock tests**
- **Fan tests**
- **Cover tests**