HDHomeRun Device Discover API

20240203-1

Low level C Library

https://download.silicondust.com/hdhomerun/libhdhomerun.tgz

https://github.com/Silicondust/libhdhomerun

The libhdhomerun library is available under a commercial license suitable for compiling directly into an app – contact Silicondust to request.

Requires C99 compliant compiler.

Example code:

See hdhomerun_discover_example.c for an example of using the discover API.

Header files:

Include hdhomerun.h to access libhdhomerun APIs.

The discover API is declared in hdhomerun_discover.h.

Source files required for discover:

All platforms	**Purpose**
hdhomerun_discover.c	Discover API
hdhomerun_debug.c	Internal debug logging
hdhomerun_pkt.c	Internal packet handling
hdhomerun_sock.c	Internal socket handling (common code)
Linux / Android	
hdhomerun_os_posix.c	Internal OS wrapper functions
hdhomerun_sock_posix.c	Internal socket handling
hdhomerun_sock_netlink.c	Internal IP address detection
Mac / BSD	
hdhomerun_os_posix.c	Internal OS wrapper functions
hdhomerun_sock_posix.c	Internal socket handling
hdhomerun_sock_getifaddrs.c	Internal IP address detection
Windows	
hdhomerun_os_windows.c	Internal OS wrapper functions
hdhomerun_sock_windows.c	Internal IP address detection & socket handling

hdhomerun_discover_create()

```
struct hdhomerun_discover_t *hdhomerun_discover_create(struct hdhomerun_debug_t *dbg);
```

Creates a discover object for managing / executing the discover process. The object may be maintained across the lifespan of the app or may be created and destroyed for each discover.

If the app polls discover often the discover object should be maintained and reused to avoid excessive cycling through local port numbers.

Parameters:

- **dbg**: debug object for debug logging or NULL if not using logging.

Returns:

- Discover object. Free using hdhomerun_discover_destroy().

hdhomerun_discover_destroy()

```
void hdhomerun_discover_destroy(struct hdhomerun_discover_t *ds);
```

Frees all resources associated with the given discover object.

Parameters:

- **ds**: discover object to free.

Returns:

- None.

hdhomerun_discover2_find_devices_broadcast()

```
int hdhomerun_discover2_find_devices_broadcast(struct hdhomerun_discover_t *ds, uint32_t flags,
uint32_t const device_types[], size_t device_types_count);
```

Execute discover process to find devices of the specified type(s) on the local network(s). Results from any prior discover are wiped.

This is the primary API to discover devices.

Execution time is 400ms.

Parameters:

- **ds**: discover object.
- **flags**: specify which IP types to use for discover. Multiple flags can be specified by OR'ing together:
    - HDHOMERUN_DISCOVER_FLAGS_IPV4_GENERAL
        - Discover using all local IPv4 addresses (excluding localhost).
    - HDHOMERUN_DISCOVER_FLAGS_IPV4_LOCALHOST
        - Discover via IPv4 localhost. Specify if the local system can support running the HDHomeRun record engine for storing recordings on the local system.
    - HDHOMERUN_DISCOVER_FLAGS_IPV6_GENERAL
        - Discover using all global and site-local IPv6 addresses.
    - HDHOMERUN_DISCOVER_FLAGS_IPV6_LINKLOCAL
        - Discover using all link-local IPv6 addresses. Only specify if the application supports tracking IPv6 scope IDs.
    - HDHOMERUN_DISCOVER_FLAGS_IPV6_LOCALHOST
        - Discover via IPv6 localhost. Specify if the local system can support running the HDHomeRun record engine for storing recordings on the local system.
- **device_types**: array of uint32_t values specifying the types of devices to detect. Values:
    - HDHOMERUN_DEVICE_TYPE_TUNER
        - Detect HDHomeRun tuner devices.
    - HDHOMERUN_DEVICE_TYPE_STORAGE
        - Detect HDHomeRun record engines.
- **device_types_count**: number of entries in the **device_types** array provided.

Returns:

- 1 if one or more devices were found.
- 0 if the discover process completed without error but no devices were found.
- -1 if the discover process failed.

Notes:

Applications that support the complete HDHomeRun DVR system should pass in a 2 element **device_type** array indicating both TUNER and STORAGE.

Applications that support live TV or don’t use the HDHomeRun DVR system should pass in a 1 element **device_type** array indicating TUNER.

HDHomeRun tuner models with a hard drive present respond as both TUNER and STORAGE and report both TUNER and STORAGE types in the results.

hdhomerun_discover2_find_devices_targeted()

```
int hdhomerun_discover2_find_devices_targeted(struct hdhomerun_discover_t *ds, const struct sockaddr
*target_addr, const uint32_t device_types[], size_t device_types_count);
```

Execute discover process to get information about the device where the IP address is known. Results from any prior discover are wiped.

Execution time can be up to 400ms if there is no response.

Parameters:

- **ds**: discover object.
- **target_addr**: IP address to send discover request to.
    - IPv4 and IPv6 addresses supported.
    - IPv6 link-local – the Scope ID is used if specified, auto-detected if 0.
    - IPv6 non link-local – set Scope ID to 0.
- **device_types**: array of uint32_t values specifying the types of devices to detect. See hdhomerun_discover2_find_devices_broadcast() for the description.
- **device_types_count**: number of entries in the **device_types** array provided.

Returns:

- 1 if the device responded successfully.
- 0 if the discover process completed without error but the device did not respond.
- -1 if the discover process failed.

hdhomerun_discover2_find_device_id_broadcast()

```
int hdhomerun_discover2_find_device_id_broadcast(struct hdhomerun_discover_t *ds, uint32_t flags,
uint32_t device_id);
```

Execute discover process to locate the specified device by Device ID (serial number) on the local network(s). Results from any prior discover are wiped.

Execution time can be up to 400ms if the device is not detected.

Parameters:

- **ds**: discover object.
- **flags**: specify which IP types to use for discover. See hdhomerun_discover2_find_devices_broadcast() for the description.
- **device_id**: uint32_t device id (serial number) of the device to locate.

Returns:

- 1 if the specified device was successfully located.
- 0 if the discover process completed without error but the specified device could not be located.
- -1 if the discover process failed.

hdhomerun_discover2_find_device_id_targeted()

```
int hdhomerun_discover2_find_device_id_targeted(struct hdhomerun_discover_t *ds, const struct sockaddr
*target_addr, uint32_t device_id);
```

Execute discover process to get information about the device where both the Device ID (serial number) and IP address are known.

Execution time can be up to 400ms if there is no response.

Parameters:

- **ds**: discover object.
- **flags**: specify which IP types to use for discover. See hdhomerun_discover2_find_devices_broadcast() for the description.
- **target_addr**: IP address to send discover request to.
  - IPv4 and IPv6 addresses supported.
  - IPv6 link-local – the Scope ID is used if specified, or auto-detected if 0.
  - IPv6 non link-local – set Scope ID to 0.
- **device_id**: uint32_t device id (serial number) of the device to locate.

Returns:

- 1 if the specified device responded successfully.
- 0 if the discover process completed without error but the specified device did not respond.
- -1 if the discover process failed.

hdhomerun_discover2_iter_device_first()

```
struct hdhomerun_discover2_device_t *hdhomerun_discover2_iter_device_first(struct hdhomerun_discover_t
*ds);
```

Return the first device from the list of devices discovered by the most recent invocation of one of the hdhomerun_discover2_find_devices_xxx() APIs.

Parameters:

- **ds**: discover object.

Returns:

- Device object. Resource maintained by the ds discover object – do not store or free.
- NULL is no devices were found.

hdhomerun_discover2_iter_device_next()

```
struct hdhomerun_discover2_device_t *hdhomerun_discover2_iter_device_next(struct
hdhomerun_discover2_device_t *device);
```

Return the next device from the list of devices discovered by the most recent invocation of one of the hdhomerun_discover2_find_devices_xxx() APIs.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or previous call to hdhomerun_discover2_iter_device_next().

Returns:

- Device object. Resource maintained by the ds discover object – do not store or free.
- NULL when the end of the list has been reached.

hdhomerun_discover2_device_is_legacy()

```
bool hdhomerun_discover2_device_is_legacy(struct hdhomerun_discover2_device_t *device);
```

Check if the detected device is a legacy tuner lacking channel management and http streaming support.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().

Returns:

- True if the device is a legacy tuner.

hdhomerun_discover2_device_is_type()

```
bool hdhomerun_discover2_device_is_type(struct hdhomerun_discover2_device_t *device, uint32_t
device_type);
```

Check if the detected device is of a given type.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().
- **device_type**: type to query:
  - HDHOMERUN_DEVICE_TYPE_TUNER
    - HDHomeRun tuner device.
  - HDHOMERUN_DEVICE_TYPE_STORAGE
    - HDHomeRun record engine.

Returns:

- True if the device is of the type specified by the **device_type** parameter.

hdhomerun_discover2_device_get_device_id()

```
uint32_t hdhomerun_discover2_device_get_device_id(struct hdhomerun_discover2_device_t *device);
```

Return the 32-bit Device ID (serial number) of the device if the device is HDHomeRun hardware.

Devices of type TUNER are hardware products and always have a Device ID.

Devices of type STORAGE may be a hardware product with a Device ID or may be a software product with no Device ID (such as the HDHomeRun DVR record engine installed on a PC or NAS). To uniquely identify devices of type STORAGE use the alternative hdhomerun_discover2_device_get_storage_id() API.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().

Returns:

- 32-bit Device ID (serial number) of the HDHomeRun hardware.
- 0 if the device does not have a Device ID, for example a DVR record engine installed on a PC or NAS.

hdhomerun_discover2_device_get_storage_id()

```
const char *hdhomerun_discover2_device_get_device_auth(struct hdhomerun_discover2_device_t *device);
```

Return the UUID string Storage ID of the device.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().

Returns:

- UUID string Storage ID. The returned result must be copied if the value needs to be accessed by the app beyond the scope of the current discover action.
- NULL if the device does not have a Storage ID, for example a tuner without storage capabilities.

hdhomerun_discover2_device_get_tuner_count()

```
uint8_t hdhomerun_discover2_device_get_tuner_count(struct hdhomerun_discover2_device_t *device);
```

Return the number of tuners the device has.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().

Returns:

- The number of tuners the device has.
- 0 if the device does not have tuners.

hdhomerun_discover2_device_get_device_auth()

```
const char *hdhomerun_discover2_device_get_device_auth(struct hdhomerun_discover2_device_t *device);
```

The DeviceAuth string reported by the device. Used to access HDHomeRun cloud APIs.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().

Returns:

- The DeviceAuth string reported by the device. The returned result must be copied if the value needs to be accessed by the app beyond the scope of the current discover action.
- NULL if the device did not report a DeviceAuth string during discovery.

hdhomerun_discover2_iter_device_if_first()

```
struct hdhomerun_discover2_device_if_t *hdhomerun_discover2_iter_device_if_first(struct
hdhomerun_discover2_device_t *device);
```

A device can have multiple IP addresses, for example IPv4, IPv6 global, and IPv6 link-local. Use hdhomerun_discover2_iter_device_if_first() to obtain the first device-if object (IP address object) in the list associated with a device. This object is used to query the IP address and associated URLs.

A typical application only needs to get the first device-if object and does not need to iterate through the list of device-if objects.

Parameters:

- **device**: device object from hdhomerun_discover2_iter_device_first() or hdhomerun_discover2_iter_device_next().

Returns:

- Device-IF object. Resource maintained by the ds discover object – do not store or free. A device-if object is always present and cannot be NULL.

hdhomerun_discover2_iter_device_if_next()

```
struct hdhomerun_discover2_device_if_t *hdhomerun_discover2_iter_device_if_next(struct
hdhomerun_discover2_device_if_t *device_if)
```

Return the next device-if object (IP address object) in the list associated with a device. This object is used to query the IP address and associated URLs.

A typical application only needs to get the first device-if object and does not need to iterate through the list of device-if objects.

Parameters:

- **device_if**: device-if object from hdhomerun_discover2_iter_device_if_first() or previous call to hdhomerun_discover2_iter_device_if_next().

Returns:

- Device-IF object. Resource maintained by the ds discover object – do not store or free.
- NULL when the end of the list has been reached.

hdhomerun_discover2_device_if_get_ip_addr()

```
void hdhomerun_discover2_device_if_get_ip_addr(struct hdhomerun_discover2_device_if_t *device_if,
struct sockaddr_storage *ip_addr);
```

Return the IP address of the device.

Parameters:

- **device_if**: device-if object from hdhomerun_discover2_iter_device_if_first() or hdhomerun_discover2_iter_device_if_next().
- **ip_addr**: sockaddr_storage struct where the IP address is returned.

Returns:

- None.

hdhomerun_discover2_device_if_get_base_url()

```
const char *hdhomerun_discover2_device_if_get_base_url(struct hdhomerun_discover2_device_if_t
*device_if);
```

Return the base URL of the device as a string.

Parameters:

- **device_if**: device-if object from hdhomerun_discover2_iter_device_if_first() or hdhomerun_discover2_iter_device_if_next().

Returns:

- Base URL as a string. The returned result must be copied if the value needs to be accessed by the app beyond the scope of the current discover action. The Base URL is always present and cannot be NULL.

hdhomerun_discover2_device_if_get_lineup_url()

```
const char *hdhomerun_discover2_device_if_get_lineup_url(struct hdhomerun_discover2_device_if_t
*device_if);
```

Return the lineup URL (lineup.json) of the device as a string. Only present for devices of type TUNER.

Parameters:

- **device_if**: device-if object from hdhomerun_discover2_iter_device_if_first() or hdhomerun_discover2_iter_device_if_next().

Returns:

- Lineup URL as a string. The returned result must be copied if the value needs to be accessed by the app beyond the scope of the current discover action.
- NULL if the device is not type TUNER.

hdhomerun_discover2_device_if_get_storage_url()

```
const char *hdhomerun_discover2_device_if_get_storage_url(struct hdhomerun_discover2_device_if_t
*device_if);
```

Return the storage URL (recorded_files.json) of the device as a string. Only present for devices of type STORAGE.

Parameters:

- **device_if**: device-if object from hdhomerun_discover2_iter_device_if_first() or hdhomerun_discover2_iter_device_if_next().

Returns:

- Storage URL as a string. The returned result must be copied if the value needs to be accessed by the app beyond the scope of the current discover action.
- NULL if the device is not type STORAGE.