

# Booting Quest on Intel Galileo

## 1 Things You Need

- An Intel Galileo Board with the switching power supply.



Figure 1: Intel Galileo Board

- A DB9 female to 3.5mm serial cable.
- A microSD card less than 32GB.
- A USB 2.0 A-Male to Mini-B cable.

## 2 Upgrade the Firmware

Quest is booted by `grub.legacy.efi` on Galileo. To make grub display boot menu correctly, we need to upgrade the board's firmware.

- Download Arduino IDE customized for Galileo board from this link. Extract the downloaded package.
- Connect the 5V power cable to Galileo board. Wait for the board to boot.
- Connect USB cable to the USB Client Port (closest to the Ethernet) and to a PC.
- Make sure the microSD card has not been inserted yet.
- Enter the directory where Arduino IDE was extracted to. Launch the IDE by command: **sudo ./arduino**
- Select the board via **Tool - Board - Intel Galileo**
- Select the port via **Tool - Serial Port - /dev/ttyAMC0**
- Select **Help - Firmware Upgrade**.
- The upgrade progress takes about 6 minutes. When the upgrade completes, a message is displayed stating **Target Firmware upgraded successfully**.

**The official tutorial:** <https://communities.intel.com/docs/DOC-22872>

### 3 Build the Customized grub\_legacy\_efi

The original grub on Galileo doesn't support multiboot feature so that it is unable to boot Quest. A customized grub is needed.

- Clone the Support repo at <https://github.com/QuestOS/Support.git>
- Under the build\_quest\_grub directory, execute build\_quest\_grub.sh and it will build Grub automatically.
- The executable we need is grub\_legacy\_efi/efi/grub.efi

## 4 Build Quest

- Follow the docker toolchain document <sup>1</sup> to set up the Quest building environment.
- Clone the Quest repo from <https://github.com/QuestOS/quest.git>
- Checkout the galileo branch.
- In the kernel directory, do 'make config.mk'.
- Modify the newly generated config.mk file to disable SMP, logger, multiprocessor, VMX, real-time USB, sandbox and FPU. Enable MMIO32. Listing 1 is the suggested configuration:

Listing 1: Suggested config.mk for Galileo

```
# Kernel configuration

# Compiler optimization level
OPT = 0

# Disable SMP
CFG += -DNO_SMP

# Disable "logger" thread
CFG += -DNO_LOGGER

# Disable ACPI support
# CFG += -DNO_ACPI

# Disable Intel Multiprocessor Specification parsing
CFG += -DNO_INTEL_MPS

# Use VMX-based virtual machines for isolation
# CFG += -DUSE_VMX

# Use PL2303 usb-serial converter for serial output
# CFG += -DUSE_PL2303

# Share UHCI USB host controller driver
# CFG += -DSHARED_UHCI

# USB provides real-time guarantees for bulk and control transactions
# CFG += -DUSB_REALTIME_ASYNC

# Use USB Migration for cross machine migration
# CFG += -DUSB_MIGRATION
```

---

<sup>1</sup>Available on the Quest OS Further Information webpage

```

# Change to use different allocator (default: tlsf)
KMALLOC = tlsf
#KMALLOC = pow2

# Read the entire file into memory on the call to open, makes
# subsequent read calls much faster but must store the entire file in
# the kernel heap
# CFG += -DONE_FILE_READ

# Enable Linux sandbox
# CFG += -DUSE_LINUX_SANDBOX

# Enable Quest-V multi-kernel without VMX enabled
# CFG += -DQUESTV_NO_VMX

# Enable Serial MMI032 driver
# This will disable port based RS232
# Enable this flag for embedded platforms such as Galileo board
CFG += -DSERIAL_MMI032

# Disable FPU
CFG += -DNO_FPU

```

- In Quest's top-level directory, do './use-docker' and you will be directed to a new shell.
- In the new shell, go to the /quest\_src directory. Now do 'make', 'make sysprogs' and 'make ramdisk.img'.
- Quest kernel is \$(Quest\_top)/kernel/quest after building
- It's the \$(Quest\_top)/sysprogs/shell we need after bulding sysprogs
- Ramdisk we get is \$(Quest\_top)/ramdisk.img

## 5 Install Grub and Quest to MicroSD Card

- Create a partition on your MicroSD and format it as FAT32.
- Copy grub.efi (section 3), quest (section 4), shell (section 4) and ramdisk.img (section 4) to the FAT32 partition and make sure the old version of these files, if exist, are replaced. (In Linux, use -f option for cp command).
- Create a boot/grub directory in the partition's top-level directory

- Copy the grub.conf file in <https://github.com/QuestOS/Support.git> to the /boot/grub directory
- Safely remove MicroSD card

**Note:** please refer to *Intel Quark SoC X1000 Board Support Package (BSP) Build and Software User Guide* for booting Linux from MicroSD

## 6 Boot Quest

- Power off Galileo board and plug off the USB cable.
- Plug in the serial cable and insert your MicroSD card into the card slot on Galileo.
- Launch minicom (may need sudo privilege) and setup serial port.
  - Type Ctrl+Z then O to open configuration menu.
  - Go to **Serial port setup** entry.
  - Set **Serial Device** to your serial device. You can use dmesg to find out the serial device name and number you are using.
  - Set **Bps/Par/Bits** to 115200 8N1
  - Set both flow control to No
  - Listing 2 is the suggested configuration.

Listing 2: Suggested configuration for serial port

```

+-----+
| A -   Serial Device       : /dev/ttyUSB0 |
| B - Lockfile Location    : /var/lock    |
| C -   Callin Program     :              |
| D -   Callout Program    :              |
| E -   Bps/Par/Bits       : 115200 8N1  |
| F - Hardware Flow Control : No         |
| G - Software Flow Control : No         |
|                                     |
|           Change which setting?       |
+-----+

```

- Power on the board.

- A boot menu will be displayed in minicom. This is the old version grub in SPI flash.
- Press `c` to exit to grub command line and enter command: `quit`
- A window will pop up and select **UEFI Internal Shell** (Press down arrow twice) to drop to EFI shell.
- In EFI shell, select your FAT32 partition. Say it is your first partition on MicroSD, enter `fs0:` (colon is needed).
- Launch the customized grub by command: `grub.efi`
- In boot menu, select Quest entry to boot.