

MYD-LD25X

RT-Linux Application Note



File status: [] Draft [✓] Release	FILE ID:	MYIR-MYD-LD25X-SW-AN03-EN-L6.1.82
	VERSION:	V1.0[Doc]
	AUTHOR:	MSW0192
	RELEASE:	2024-11-21
	UPDATED:	2024-11-21



Revision History

VERSION	AUTHOR	PARTICIPANT	DATE	DESCRIPTION
V1.0[Doc]	MSW0192	MSW0041	2024-11-21	Official Release



CONTENT

Revision History	- 1 -
CONTENT	- 2 -
1. Overview	- 3 -
2. Hardware Resources	- 4 -
3. Software Resources	- 4 -
4. Environment Setup	- 5 -
5. Build the Image	- 6 -
5.1. Obtain the Source Code	- 6 -
5.2. Quick Compilation of the Development Board Image	- 9 -
6. Real-Time Testing	- 11 -
7. References	- 13 -
Appendix A	- 14 -

1. Overview

RT-Linux (Real-Time Linux) is a real-time operating system extension developed to address the limitations of the standard Linux kernel in handling real-time tasks. The core goal of the RT-Linux patch set is to enable the Linux kernel to provide better response times and precision in real-time applications, making it especially suitable for fields like industrial automation, embedded systems, and robotics control.

The scheduling mechanism used by the standard Linux kernel is not suitable for real-time tasks, and may lead to unpredictable delays in task execution. The RT-Linux patch introduces hard real-time features, allowing the kernel to respond quickly when real-time tasks arrive. It achieves "full preemption" of the kernel, meaning that it allows real-time tasks with higher priority to interrupt and switch from any running kernel code, significantly reducing task scheduling latency.

Another key feature of the RT-Linux patch is the optimization of interrupt handling. The patch reduces interrupt disable times and optimizes hardware interrupts, enabling the kernel to quickly handle external events and ensure timely execution of real-time tasks. Furthermore, RT-Linux improves the synchronization mechanisms between the kernel and user space, enhancing the real-time task processing capability in multi-core systems.

This chapter will use Yocto to build an RT-Linux real-time system image, using kernel version 6.1.82 and its RT-Linux patches.



2. Hardware Resources

- MYD-LD25X Development Board

3. Software Resources

- Ubuntu 20.04 64-bit (physical machine or virtual machine)
- MYD-LD25X Linux Software Development Guide

4. Environment Setup

To compile the RT-Linux image, the Yocto environment needs to be set up first. The following preparations are required for the Ubuntu system:

- **Install Necessary Tools**

This section is very important. Please ensure that the host environment correctly executes the following operations and reboot the system after completing these steps.

```
$ sudo apt-get update
$ sudo apt-get install -y gawk wget git-core diffstat \
unzip texinfo gcc-multilib build-essential chrpath socat libsdl1.2-dev \
xterm sed cvs subversion coreutils texi2html docbook-utils python-pysqlite2 \
help2man make gcc g++ desktop-file-utils libgl1-mesa-dev libglu1-mesa-dev \
mercurial autoconf automake groff curl lzip asciidoc u-boot-tools cpio \
sudo locales bc libncurses5-dev screen flex bison vim-tiny \
device-tree-compiler xvfb libgtk2.0-dev libssl-dev net-tools libyaml-dev \
rsync liblz4-tool zstd python3-pip git-lfs iputils-ping jq
$ sudo rm -rf /var/lib/apt/lists/*
$ sudo ln -s /usr/bin/python3 /usr/bin/python
$ sudo sed -i -e 's/# en_US.UTF-8 UTF-8/en_US.UTF-8 UTF-8/' /etc/locale.gen
$ echo 'LANG="en_US.UTF-8" | sudo tee /etc/default/locale > /dev/null
$ sudo dpkg-reconfigure --frontend=noninteractive locales
$ sudo update-locale LANG=en_US.UTF-8
$ sudo pip3 install pyusb usb crypto ecdsa crcmod tqdm pycryptodome pycryptodomex pyelftools
```



5. Build the Image

5.1. Obtain the Source Code

We provide two ways to obtain the source code:

Directly download the compressed source package from the 04-Sources directory in the release materials provided by MYIR. Or use repo to fetch the source code from GitHub, which is updated in real-time.

Please choose either Section 5.1.1 or Section 5.1.2 based on your needs for the build process. Since Yocto requires downloading all software packages in the filesystem locally before building, to speed up the build process, MYD-LD25X has already packaged the relevant software, which can be directly extracted for use, reducing the time spent on redundant downloads. The release materials can be accessed through MYIR's Developer Center.

5.1.1. Obtain the Source Code from the Release Materials

The source package is located in the 04-Sources directory under MYD-LD25X-Distribution-L6.1.82-V*.tar.gz. Copy the compressed package to the user-designated directory, for example, extract the source package to the working directory MYD-LD25X::

```
$ mkdir -p MYD-LD25X
$ cd MYD-LD25X
$ tar -xvf MYD-LD25X-Distribution-L6.1.82-V*.tar.gz \
-C MYD-LD25X
$ cd MYD-LD25X/MYD-LD25X-Distribution-L6.1.82-V*/MYD-LD25X-Yocto-mickle-
dore-V*
```

If you have already obtained the Yocto source code as described in this section, you can skip Section 5.2.2.

5.1.2. Obtain from GitHub

Currently, both the BSP source code and Yocto source code for the MYD-LD25X development board are hosted on GitHub and will continue to be updated over time. Please refer to the “MYD-LD25X SDK Release Note” for the repository URL. Users can use `repo` to fetch and sync the code from GitHub. The specific steps are as follows:

```
$ mkdir -p ~/bin
$ curl https://mirrors.tuna.tsinghua.edu.cn/git/git-repo > ~/bin/repo
$ chmod a+x ~/bin/repo
$ export PATH=~/bin:${PATH}
$ export REPO_URL='https://mirrors.tuna.tsinghua.edu.cn/git/git-repo/'
$ mkdir -p MYD-LD25X-yocto
$ cd MYD-LD25X-yocto
$ repo init -u https://github.com/MYiR-Dev/myir-st-manifest.git \
--no-clone-bundle --depth=1 -m myir-stm32mp2-6.1.82-1.0.0.xml \
-b myd-ld25x-v24.06.26-mickledore
$ repo sync
$ ls
layers
```

After successfully syncing the code, you will obtain the same directory structure under the MYD-LD25X directory as you would with the MYD-LD25X-Yocto-mickledore-V* source code.

5.1.3. Obtain the meta-myr-x-linux-rt Layer

After completing the above steps to obtain the source code, follow the instructions below to obtain the meta-myr-x-linux-rt layer in the layers/meta-myr directory:

- **Enter the layers/meta-myr directory:**

```
$ cd layers/meta-myr
```

- **Obtain meta-myr-x-linux-rt**

- Fetch it from GitHub:

```
$ git clone https://github.com/MYiR-Dev/meta-myr-x-linux-rt.git \
-b develop-v24.06.26-mickledore
```

- Download the meta-myr-x-linux-rt.tar.gz file from the 04-Sources/RT-Linux directory:

```
$ mv ~/downloads/04-Sources/RT-Linux/meta-myr-x-linux-rt.tar.gz ./
$ tar xf meta-myr-x-linux-rt.tar.gz
```

Choose one of the above methods to obtain the source code. Once the process is complete, the layers/meta-myr directory will appear as shown below:

```
$ ls -l
meta-myr-openstlinux
meta-myr-stm32mp
meta-myr-stm32mp-addons
meta-myr-stm32mp-tsn-acm
meta-myr-stm32mp-tsn-swch
meta-myr-x-linux-rt
scripts
```

5.2. Quick Compilation of the Development Board Image

5.2.1. Execute the Environment Variable Setup Script

Before using the Yocto project to build the system, it is necessary to set the corresponding environment variables. Therefore, each time you compile Yocto in a new terminal window, follow these steps to execute the envsetup.sh script located in the layers/meta-myr/scripts directory of the Yocto source.

Return to the Yocto directory and execute the following command to set up the build environment:

```
$ cd MYD-LD25X-yocto
$ DISTRO=openstlinux-rt MACHINE=myd-ld25x-rt source \
layers/meta-myr/scripts/envsetup.sh
```

You will be prompted to press 'y' to agree to the EULA (End User License Agreement) when executing the setup script. After the configuration script completes, it will automatically enter the generated build-openstlinuxrt-myd-ld25x-rt directory.

Note: Yocto must be compiled using a regular user account, not the root user.

5.2.2. Build the Image

In this section, we will build the myir-image-core-rt image. The differences between myir-image-core and myir-image-core-rt images are shown below:

Table 5-1. Differences Between core and core-rt Images

	myir-image-core	myir-image-core-rt
Init system	Systemd	SysV Init
Weston	Have Weston	Don't have Weston



Netdata	netdata 1.36.1	Don' t have netdata
OPTEE HSI cyclic calibration	Open	Close

After completing the environment setup, you can directly execute the following command to build the myir-image-core-rt image:

```
$ bitbake myir-image-core-rt
```

5.2.3. Flash the Image

After the build is complete, the image will be located in the tmp-glibc/deploy/images/myd-ld25x-rt/ directory. Follow the instructions in the Mass Production Guide to flash the myir-image-core-rt image onto the development board.

You can refer to Chapter 2 of the MYD-LD25X Mass Production Guide to choose the appropriate .tsv file from tmp-glibc/deploy/images/myd-ld25x-rt/flashlayout_myir-image-core-rt/optee/ (either the 1GB or 2GB FlashLayout_emmc_myb-stm32mp257x-1GB-optee.tsv file) and use STM32CubeProgrammer to flash the image onto the development board.

6. Real-Time Testing

This article uses the cyclicttest program to test the real-time performance of the system. The principle of cyclicttest is as follows:

A master process is started, which then spawns a specified number of real-time processes with specified priorities. Each real-time process sets a timer that periodically wakes it up (the delay is measured from when the timer overflows, triggers an interrupt, and enters the ISR to call wake_up_process(), until the process can actually be scheduled). Once the real-time process is running, it will obtain the current system time, subtract the time at which it went to sleep, and the sleep time itself to compute the delay. This value is then passed to the master process via shared memory for statistical analysis. This process repeats in cycles, and eventually, the master process outputs the results.

6.1. Cyclicttest Testing

Before testing with cyclicttest, execute the following command to run the stress-ng tool and simulate a load environment:

```
# stress-ng -c 24 --cpu-method fft --timerfd-freq 1000000 -t 24h &
```

Then, run the following command to start the cyclicttest test:

```
# cyclicttest --mlockall --smp --priority=98 --interval=1000 --distance=0 -D72h
```

6.2. Real-Time Performance Comparison

By reading this section, you can observe the changes and improvements in system real-time performance before and after applying the real-time patch. It will also provide you with a reference record.

cyclicttest results before applying the real-time patch:

```
root@myd-1d25x-rt:~# cyclicttest --mlockall --smp --priority=98 --interval=1000 --distance=0 -D72h
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 21.84 8.70 3.22 25/198 674
T: 0 ( 673) P:98 I:1000 C: 424 Min: 21 Act: 60 Avg: 47 Max: 60
T: 1 ( 674) P:98 I:1000 C: 397 Min: 20 Act: 42 Avg: 37 Max: 144
```

Figure 6-1. Cyclicttest Results Before Applying the Real-Time Patch

Cyclicttest Results After Applying the Real-Time Patch:

```
root@myd-1d25x-rt:~# cyclicttest --mlockall --smp --priority=98 --interval=1000 --distance=0 -D72h
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 24.17 22.33 13.22 25/184 761
T: 0 ( 760) P:98 I:1000 C: 1146 Min: 13 Act: 19 Avg: 18 Max: 39
T: 1 ( 761) P:98 I:1000 C: 974 Min: 15 Act: 24 Avg: 19 Max: 30
```

Figure 6-2. Cyclicttest Results After Applying the Real-Time Patch

Cyclicttest Long-Term Test Results After Applying the Real-Time Patch:

```
root@myd-1d25x-rt:~# cyclicttest --mlockall --smp --priority=98 --interval=1000 --distance=0 -D72h
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 21.84 8.70 3.22 25/198 674
T: 0 ( 673) P:98 I:1000 C:2592000000 Min: 12 Act: 15 Avg: 18 Max: 39
T: 1 ( 674) P:98 I:1000 C:2592000000 Min: 13 Act: 20 Avg: 19 Max: 40
```

Figure 6-3. Long-Term Test Results



7. References

- **ST X-Linux-Rt Wiki**

https://wiki.st.com/stm32mpu/wiki/X-LINUX-RT_expansion_package#

Appendix A

Warranty & Technical Support Services

MYIR Electronics Limited is a global provider of ARM hardware and software tools, design solutions for embedded applications. We support our customers in a wide range of services to accelerate your time to market.

MYIR is an ARM Connected Community Member and work closely with ARM and many semiconductor vendors. We sell products ranging from board level products such as development boards, single board computers and CPU modules to help with your evaluation, prototype, and system integration or creating your own applications. Our products are used widely in industrial control, medical devices, consumer electronic, telecommunication systems, Human Machine Interface (HMI) and more other embedded applications. MYIR has an experienced team and provides custom design services based on ARM processors to help customers make your idea a reality.

The contents below introduce to customers the warranty and technical support services provided by MYIR as well as the matters needing attention in using MYIR's products.

Service Guarantee

MYIR regards the product quality as the life of an enterprise. We strictly check and control the core board design, the procurement of components, production control, product testing, packaging, shipping and other aspects and strive to provide products with best quality to customers. We believe that only quality products and excellent services can ensure the long-term cooperation and mutual benefit.

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

Delivery Time

MYIR will always keep a certain stock for its regular products. If your order quantity is less than the amount of inventory, the delivery time would be within three days; if your order quantity is greater than the number of inventory, the delivery time would be always four to six weeks. If for any urgent delivery, we can negotiate with customer and try to supply the goods in advance.

Technical Support



MYIR has a professional technical support team. Customer can contact us by email (support@myirtech.com), we will try to reply you within 48 hours. For mass production and customized products, we will specify person to follow the case and ensure the smooth production.

After-sale Service

MYIR offers one year free technical support and after-sales maintenance service from the purchase date.

The service covers:

Technical support service

MYIR offers technical support for the hardware and software materials which have provided to customers;

- To help customers compile and run the source code we offer;
- To help customers solve problems occurred during operations if users follow the user manual documents;
- To judge whether the failure exists;
- To provide free software upgrading service.

However, the following situations are not included in the scope of our free technical support service:

- Hardware or software problems occurred during customers' own development;
- Problems occurred when customers compile or run the OS which is tailored by themselves;
- Problems occurred during customers' own applications development;
- Problems occurred during the modification of MYIR's software source code.

After-sales maintenance service

The products except LCD, which are not used properly, will take the twelve months free maintenance service since the purchase date. But following situations are not included in the scope of our free maintenance service:

- The warranty period is expired;
- The customer cannot provide proof-of-purchase or the product has no serial number;
- The customer has not followed the instruction of the manual which has caused the damage the product;
- Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads the defects of appearance/function;
- Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which have caused the damage of the products or defects of appearance;
- Due to unauthorized weld or dismantle parts or repair the products which has caused the damage of the products or defects of appearance;



- Due to unauthorized installation of the software, system or incorrect configuration or computer virus which has caused the damage of products.

Warm tips

1. MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.
2. Please do not use finger nails or hard sharp object to touch the surface of the LCD.
3. MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.
4. Do not clean the surface of the screen with chemicals.
5. Please read through the product user manual before you using MYIR's products.
6. For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR's support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

Maintenance period and charges

- MYIR will test the products within three days after receipt of the returned goods and inform customer the testing result. Then we will arrange shipment within one week for the repaired goods to the customer. For any special failure, we will negotiate with customers to confirm the maintenance period.
- For products within warranty period and caused by quality problem, MYIR offers free maintenance service; for products within warranty period but out of free maintenance service scope, MYIR provides maintenance service but shall charge some basic material cost; for products out of warranty period, MYIR provides maintenance service but shall charge some basic material cost and handling fee.

Shipping cost

During the warranty period, the shipping cost which delivered to MYIR should be responsible by user; MYIR will pay for the return shipping cost to users when the product is repaired. If the warranty period is expired, all the shipping cost will be responsible by users.

Products Life Cycle



MYIR will always select mainstream chips for our design, thus to ensure at least ten years continuous supply; if meeting some main chip stopping production, we will inform customers in time and assist customers with products updating and upgrading.

Value-added Services

1. MYIR provides services of driver development base on MYIR's products, like serial port, USB, Ethernet, LCD, etc.
2. MYIR provides the services of OS porting, BSP drivers' development, API software development, etc.
3. MYIR provides other products supporting services like power adapter, LCD panel, etc.
4. ODM/OEM services.

MYIR Electronics Limited

Room 04, 6th Floor, Building No.2, Fada Road,
Yunli Intelligent Park, Bantian, Longgang District.

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836

Fax: +86-755-25532724

Website: www.myirtech.com