How To Do ?

Development Environment Setup - **MA35D0**

MU20 Eric

MPU Platform Practice

Exercise 1 MA35D0 Setup

1. Development Environment Setup
2. Make an Embedded Image
3. NuWriter to Program Image
4. Run Application Code

MPU Quick Start

- Development Environment Setup

- **Environment Download:**
  1. Ubuntu OS: MA35D1 Linux Environment on VMWare
  2. VMWare player : VMware Player

- **Software Tool :**
  Programming : MA35_NuWriter

MPU Platform Practice

Exercise 1 MA35D0 Setup

1. Development Environment Setup
2. Make an Embedded Image
3. NuWriter to Program Image
4. Run Application Code

MPU Quick Start

- Make an Embedded Image

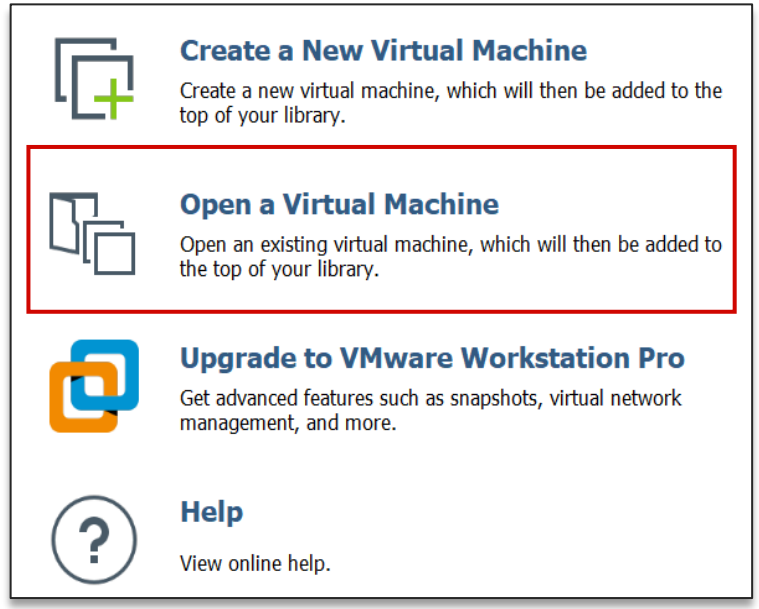
➢ Start up with VMware

- This VMware Image offers a Linux development environment for MA35H0.
- User Name: user

  Password: user

Open the development environment

1. Click *Vmware player*
2. Click "*Open a Virtual Machine" and choose* ubuntu image(unzip)
3. Enter the password to enter the linux OS



MA35D0 Quick Start

- Make an Embedded Image

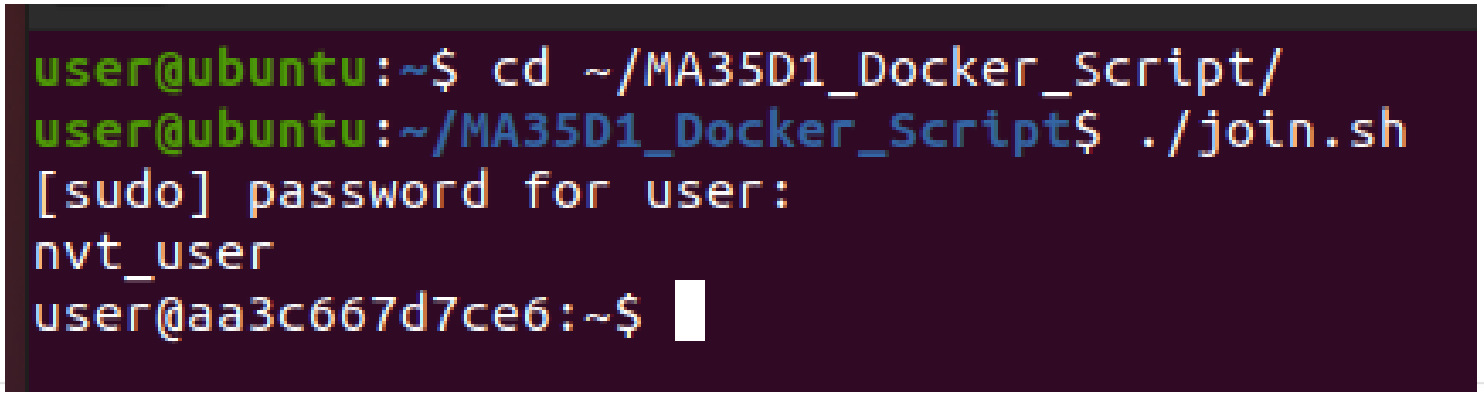
➢ Into to the Docker

Buildroot:

$ cd ~/MA35D1_Docker_Script/

$ ./join.sh

Password: user



MPU Quick Start

- Make an Embedded Image

➢ [Docker] Update Buildroot

Buildroot:

Goto MA35D1_Buildroot folder

$ cd ~/shared/buildroot/MA35D1_Buildroot/

Update Buildroot:

$ git pull

```
user@aa3c667d7ce6:~/shared/Buildroot/MA35D1_Buildroot$ git pull
Already up to date.
user@aa3c667d7ce6:~/shared/Buildroot/MA35D1_Buildroot$
```

MA35D0 Quick Start

- Make an Embedded Image

➢ [Docker] Choose your EVB config

Buildroot:

- Set configuration to the target board

$ make numaker_iot_ma35d03f80_defconfig

```
user@aa3c667d7ce6:~/shared/Buildroot/MA35D0_Buildroot_IoT$ make numaker_iot_ma35d03f80_defconfig
#
# configuration written to /home/user/shared/Buildroot/MA35D0_Buildroot_IoT/.config
#
user@aa3c667d7ce6:~/shared/Buildroot/MA35D0_Buildroot_IoT$
```

MA35D0 Quick Start

- Make an Embedded Image

➢ [Docker] Make it

Buildroot:

- start building it. (*first build need 1 hour*)

$ make

```
user@aa3c667d7ce6:~/shared/Buildroot/MA35D1_Buildroot$ make
>>>   Finalizing host directory
>>>   Finalizing target directory
mkdir -p /home/user/shared/Buildroot/MA35D1_Buildroot/output/host/etc/meson
sed -e 's%@TARGET_CROSS@%/home/user/shared/Buildroot/MA35D1_Buildroot/output/host/bin/aarch64-nuvoton-linux-gnu-%g' -e 's%@TARGET_ARCH@%aarch64%g
' -e 's%@TARGET_CPU@%cortex-a35%g' -e 's%@TARGET_ENDIAN@%little%g' -e "s%@TARGET_CFLAGS@%'-D_LARGEFILE_SOURCE', '-D_LARGEFILE64_SOURCE', '-D_FILE
_OFFSET_BITS=64', '-Os'@PKG_TARGET_CFLAGS@%g" -e "s%@TARGET_LDFLAGS@%@PKG_TARGET_CFLAGS@%g" -e "s%@TARGET_CXXFLAGS@%'-D_LARGEFILE_SOURCE', '-D_LA
RGEFILE64_SOURCE', '-D_FILE_OFFSET_BITS=64', '-Os'@PKG_TARGET_CFLAGS@%g" -e 's%@HOST_DIR@%/home/user/shared/Buildroot/MA35D1_Buildroot/output/hos
t%g' -e 's%@STAGING_DIR@%/home/user/shared/Buildroot/MA35D1_Buildroot/output/host/aarch64-nuvoton-linux-gnu/sysroot%g' -e 's%@STATIC@%false%g' pa
ckage/meson//cross-compilation.conf.in > /home/user/shared/Buildroot/MA35D1_Buildroot/output/host/etc/meson/cross-compilation.conf.in
sed -e 's%@PKG_TARGET_CFLAGS@%%g' -e 's%@PKG_TARGET_LDFLAGS@%%g' -e 's%@PKG_TARGET_CXXFLAGS@%%g' /home/user/shared/Buildroot/MA35D1_Buildroot/out
put/host/etc/meson/cross-compilation.conf.in > /home/user/shared/Buildroot/MA35D1_Buildroot/output/host/etc/meson/cross-compilation.conf
/usr/bin/sed -i -e '/# GENERIC_SERIAL$/s~^.*#~ttyS0::respawn:/sbin/getty -L  ttyS0 0 vt100 #~' /home/user/shared/Buildroot/MA35D1_Buildroot/outpu
t/target/etc/inittab
/usr/bin/sed -i -e '/^#.*-o remount,rw \/$/s~^#\+~~' /home/user/shared/Buildroot/MA35D1_Buildroot/output/target/etc/inittab
if grep -q CONFIG_ASH=y /home/user/shared/Buildroot/MA35D1_Buildroot/output/build/busybox-1.33.1/.config; then grep -qsE '^/bin/ash$' /home/user/
shared/Buildroot/MA35D1_Buildroot/output/target/etc/shells || echo "/bin/ash" >> /home/user/shared/Buildroot/MA35D1_Buildroot/output/target/etc/s
hells; fi
if grep -q CONFIG_HUSH=y /home/user/shared/Buildroot/MA35D1_Buildroot/output/build/busybox-1.33.1/.config; then grep -qsE '^/bin/hush$' /home/use
r/shared/Buildroot/MA35D1_Buildroot/output/target/etc/shells || echo "/bin/hush" >> /home/user/shared/Buildroot/MA35D1_Buildroot/output/target/et
c/shells; fi
rm -f /home/user/shared/Buildroot/MA35D1_Buildroot/output/target/usr/share/glib-2.0/schemas/*.xml /home/user/shared/Buildroot/MA35D1_Buildroot/ou
tput/target/usr/share/glib-2.0/schemas/*.dtd
/home/user/shared/Buildroot/MA35D1_Buildroot/output/host/bin/glib-compile-schemas /home/user/shared/Buildroot/MA35D1_Buildroot/output/host/aarch6
4-nuvoton-linux-gnu/sysroot/usr/share/glib-2.0/schemas --targetdir=/home/user/shared/Buildroot/MA35D1_Buildroot/output/target/usr/share/glib-2.0/
schemas
```

MA35D0 Quick Start

- Make an Embedded Image

➢ [Docker] Building Finishing

Buildroot:

- Once the build is complete, you can find the resulting image at /output/image

```
bl2.bin                                                  ma35d1-som-256m.dtb                                      tee.bin
bl2.dtb                                                  MBR.sdcard.bin                                           tee-header_v2.bin
bl31.bin                                                 nuwriter                                                 tee-pageable_v2.bin
core-image-buildroot-ma35d1-som-256m.rootfs.sdcard       pack-core-image-buildroot-ma35d1-som-256m-sdcard.bin     tee-pager_v2.bin
fip.bin                                                  rootfs.ext2                                              u-boot.bin
fip.bin-sdcard                                           rootfs.ext4                                              uboot-env.bin
header.bin                                               rootfs.tar                                               uboot-env.bin-sdcard
header-core-image-buildroot-ma35d1-som-256m-sdcard.bin   rootfs.ubi                                               uboot-env.txt
Image                                                    rootfs.ubifs                                             uboot-env.txt-sdcard
Image.dtb                                                RTP-BSP
user@aa3c667d7ce6:~/shared/Buildroot/MA35D1_Buildroot/output/images$
```

- The core-image-buildroot-ma35d0-iot-256m.rootfs.sdcard can be used to directly program to an SD card
- pack-core-image-buildroot-ma35d0-iot-256m-sdcard.bin can be used to program an SD card through NuWriter

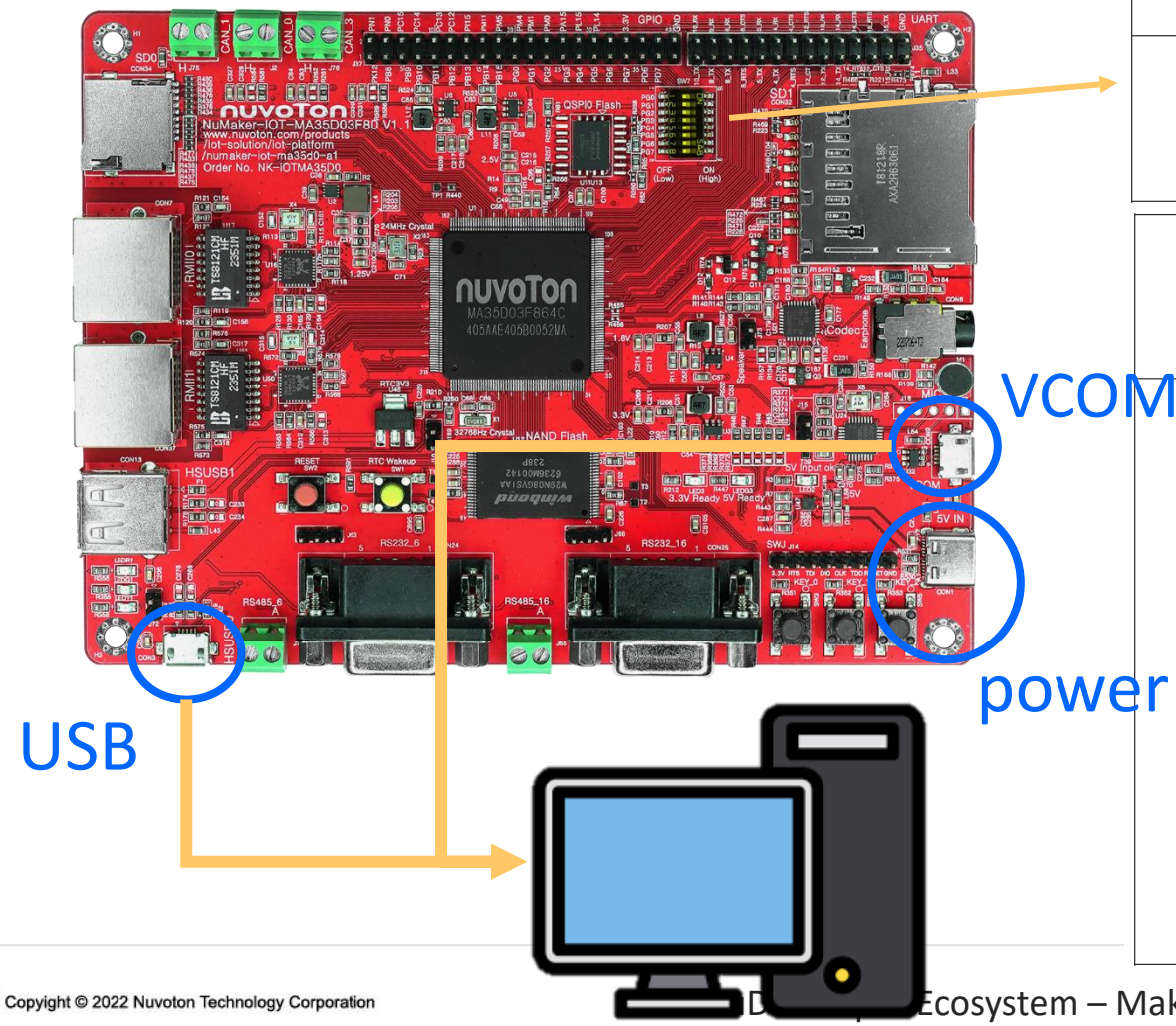
MPU Platform Practice

Exercise 1 MA35D0 Setup

1. Development Environment Setup
2. Make an Embedded Image
3. NuWriter to Program Image
4. Run Application Code

MA35D0 Quick Start

- NuWriter to Program Image



Power-on-setting

Pin	Description
PG[0]	Secure Boot Disable Bit 0 = Secure Boot Enabled. 1 = Secure Boot Disabled.
PG[3:2]	Boot Source Selection 00 = Boot from SPI Flash. 01 = Boot from SD/eMMC. 10 = Boot from NAND Flash. 11 = Boot from USB.
PG[5:4]	NAND Flash Page Size Selection 00 = Ignore. 01 = NAND Flash page size is 2 KB. 10 = NAND Flash page size is 4 KB. 11 = NAND Flash page size is 8 KB.
PG[7:6]	Miscellaneous Configuration If BTSRCSEL = 01, Boot from SD/eMMC. 00 = SD0/eMMC0 4-bit mode booting. 01 = SD1/eMMC1 4-bit mode booting. 10 = eMMC0 8-bit mode booting. 11 = eMMC1 8-bit mode booting. If BTSRCSEL = 10, the Boot from NAND Flash. 00 = Ignore. 01 = ECC is BCH T12. 10 = ECC is BCH T24. 11 = No ECC. If BTSRCSEL = 00, the Boot from SPI Flash. 00 = SPI-NAND Flash with 1-bit mode booting. 10 = SPI-NOR Flash with 1-bit mode booting. Others reserved.

Remember to USB mode

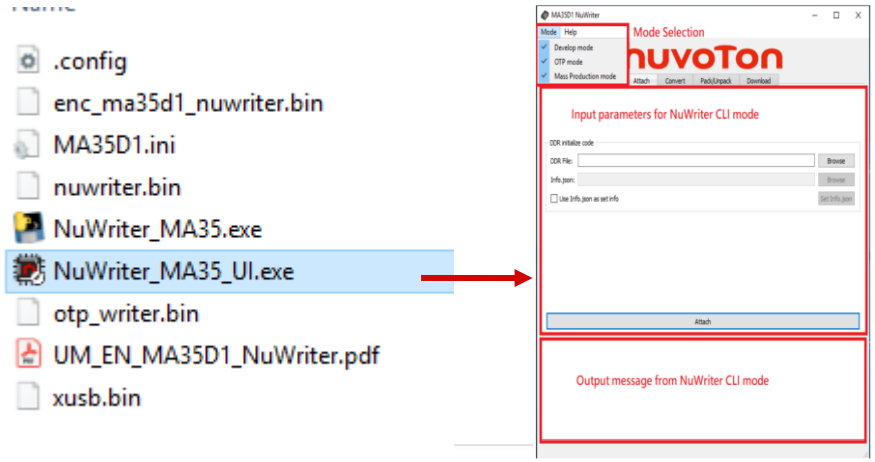
SW 1 & SW 3 & SW 4 are ON
Others sre OFF.

MA35D0 Quick Start

- NuWriter to Program Image

➢ Reset Btn for reset MA35D0

➢ Open the NuWriter

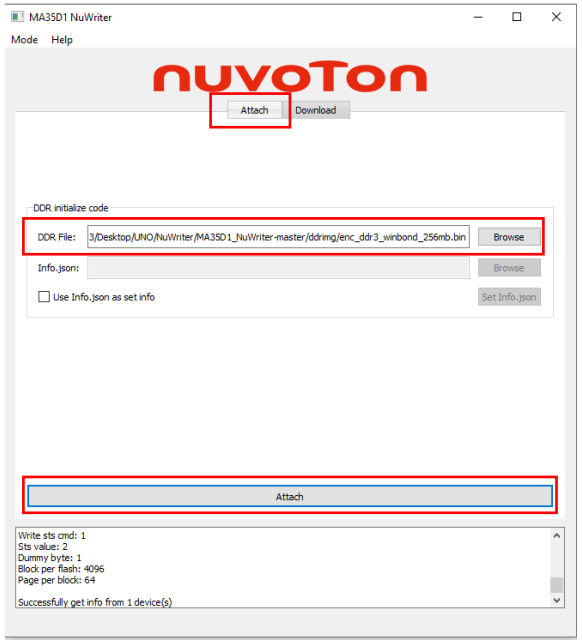


MA35D0 Quick Start

- NuWriter to Program Image

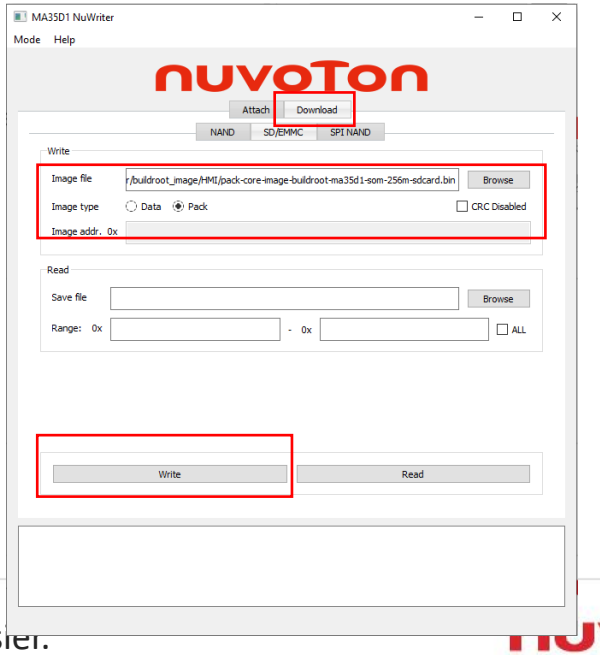
➢ First, Attach

Choose \ddrimg\MA35D03F864C.bin and Attach for DDR init.



➢ Second, Download

1. Choose Download , SD/EMMC
2. Select .bin file : pack-core-image-buildroot-ma35d0-iot-256m-sdcard.bin
3. pack tick
4. Write



MPU Platform Practice

Exercise 1 MA35D0 Setup

1. Development Environment Setup
2. Make an Embedded Image
3. NuWriter to Program Image
4. Run Application Code

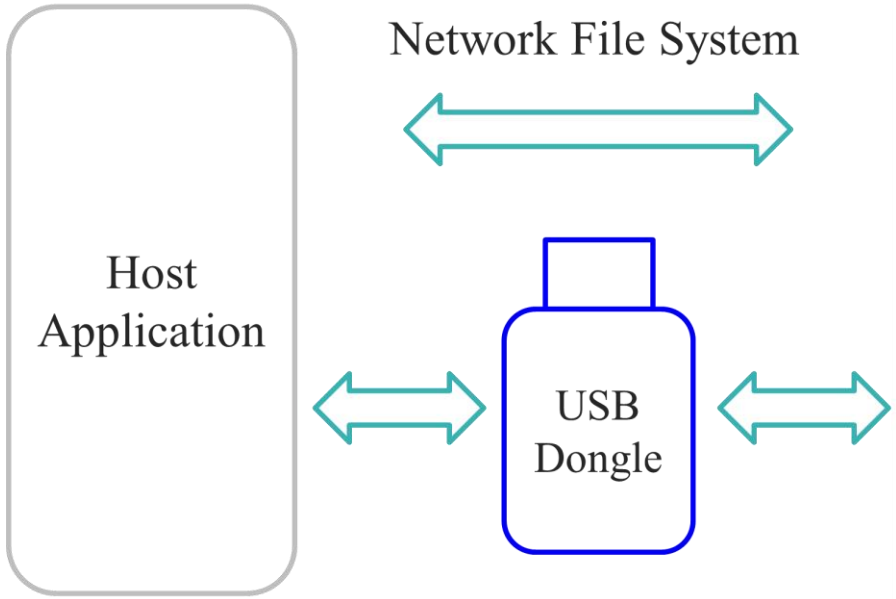
MA35D0 Quick Start

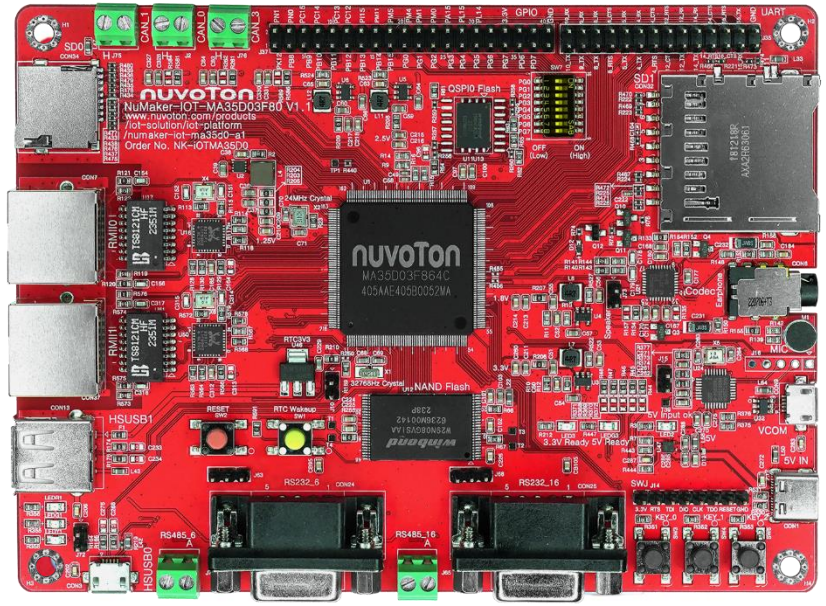
- Run Application Code

➢ Put SimpleDemoAppWizard to MA35D0

There are many ways to program application to evaluation board. Here Nuvoton demonstrates two methods.

1. Network file System Programming
2. USB Dongle Programming





Joy of innovation

NUVOTON

谢谢

謝謝

Děkuji

Bedankt

Thank you

Kiitos

Merci

Danke

Grazie

ありがとう

감사합니다

Dziękujemy

Obrigado

Спасибо

Gracias

Teşekkür ederim

Cảm ơn