

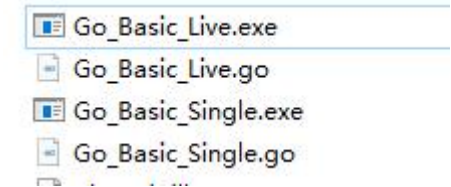
1.根据不同平台下载并安装 Go 开发环境: <https://go.dev/dl/>

File name	Kind	OS	Arch	Size	SHA256 Checksum
go1.25.0.src.tar.gz	Source			30MB	4b401e91297207bfa450ee40d4d5a93b1b531a5e438473b2a06e18e077227225
go1.25.0.darwin-amd64.tar.gz	Archive	macOS	x86-64	58MB	5b460e823037062c2307e71e8111809865116714d6f6b410597cf5075df480ef
go1.25.0.darwin-amd64.pkg	Installer	macOS	x86-64	59MB	95e836238bcefb9a71bffe43344cb35ee1f16db3aaced2f98dbac045d102db
go1.25.0.darwin-arm64.tar.gz	Archive	macOS	ARM64	55MB	544932844156d8172f7a28f77f2ae9e15a2304669b6243f633b0a0b00e0749e
go1.25.0.darwin-arm64.pkg	Installer	macOS	ARM64	56MB	202a0d8338e152eb4e9f04782429e9ba8bef31d9889272380837e4043e9d800a
go1.25.0.linux-386.tar.gz	Archive	Linux	x86	56MB	8e602d8d99bce9453b3995d20ce4ba382ce50855900a0ee5da9929df4a993a
go1.25.0.linux-amd64.tar.gz	Archive	Linux	x86-64	57MB	2852af0cb20a13139b3448992e69b868e50ed0f8a1e5940ee1de9e19a123b613
go1.25.0.linux-arm64.tar.gz	Archive	Linux	ARM64	55MB	05da75d6994a2783699815ee553bd5a9327d8b79991da36e38b66862782f54ae
go1.25.0.linux-armv6l.tar.gz	Archive	Linux	ARMv6	56MB	a5a9f8198f0f00e1e485b8e9e9e020778bf32a408a4e8873371bfce450cd09
go1.25.0.windows-386.zip	Archive	Windows	x86	63MB	df9f39db82a803af0db639a3613a36681ab7e42866b1384b3f3a1045663961a7
go1.25.0.windows-386.msi	Installer	Windows	x86	52MB	af89e0a8d2665ff122c8302bb4a3ce4a5331e4e630ddc388be1f9238adf8fa3
go1.25.0.windows-amd64.zip	Archive	Windows	x86-64	64MB	89efb4f9b30812ee083ec1770fd3d2913c14d301064f6454851428e9707d190b
go1.25.0.windows-amd64.msi	Installer	Windows	x86-64	54MB	936bd87109da515f79d80211de5bce6ebda071f2cc577f7e6af1a9e754ea34819
Other Ports ^					

2.安装完成后打开终端，输入 go env，若有输出则说明环境安装正确

```
C:\Windows\system32\cmd.exe
C:\Users\Q>go env
set AR=ar
set CC=gcc
set CGO_CFLAGS=-O2 -g
set CGO_CPPFLAGS=
set CGO_CXXFLAGS=-O2 -g
set CGO_ENABLED=1
set CGO_FFLAGS=-O2 -g
set CGO_LDFLAGS=-O2 -g
set CXX=g++
set GCCGO=gccgo
set GO111MODULE=
set GOAMD64=v1
set GOARCH=amd64
set GOAUTH=netrc
set GOBIN=
```

3.在终端上进入 Demo 程序所在文件夹，通过 go build + go 程序可以生成运行程序，也可以通过 go run + go 程序名直接运行 go 程序。



```
C:\Windows\system32\cmd.exe

D:\Workspace\Demo\Go_Demo\Go_Basic>go run Go_Basic_Single.go
Operation timed out or was aborted
Operation timed out or was aborted
C_InitQHYCCDResource() ret = 0
C_ScanQHYCCD() num = 1
C_GetQHYCCDId() ret = 0 id = QHY268M-31c44548beeff4680
C_OpenQHYCCD() handle = 0x2c91d5e6250
C_GetQHYCCDReadModeNumber() ret = 0 modeNum = 7
C_GetQHYCCDReadModeName() ret = 0 modeName = PhotoGraphic DSO
C_SetQHYCCDReadMode() read mode 0 ret = 0
C_SetQHYCCDStreamMode() stream mode 0 ret = 0
C_InitQHYCCD() ret = 0
C_SetQHYCCDDebayerOnOff() debayer false ret = 0
C_SetQHYCCDParam_Bits() 16 bits ret = 0
C_GetQHYCCDChipInfo() ret = 0 chipw = 23.612800mm chiph = 15.829600mm imagew = 6280
imageh = 4210 pixelw = 3.760000um pixelH = 3.760000um bpp = 16
C_SetQHYCCDBinMode() bin 1x1 ret = 0
C_SetQHYCCDResolution() resolution 0 x 6280 ret = 4210
C_SetQHYCCDParam_Exposure() exposure 50ms ret = 0
C_SetQHYCCDParam_Gain() gain 50 ret = 0
C_SetQHYCCDParam_Offset() offset 50 ret = 0
C_ExpQHYCCDSingleFrame() ret = 0
C_GetQHYCCDSingleFrame() ret = 0 w = 6280 h = 4210 bpp = 16 channels = 1
C_CancelQHYCCDExposingAndReadout() ret = 0
C_CloseQHYCCD() ret = 0
C_ReleaseQHYCCDResource() ret = 0

D:\Workspace\Demo\Go_Demo\Go_Basic>_
```