Lerobot添加新机械臂tutorial

> 还在clear code，看看优化完成能不能提交PR（在做梦hhh

🤗 LeRobot：以dummy为例新增机械臂的一般流程

1. `lerobot/common/robot_devices/motors/`
    a. 创建 `dummy.py` ：实现 `class DummyMotorsBus`

dummy.py

```python
class MotorsBus(Protocol):
    def motor_names(self): ...
    def set_calibration(self): ...
    def apply_calibration(self): ...
    def revert_calibration(self): ...
    def read(self): ...
    def write(self): ...
```

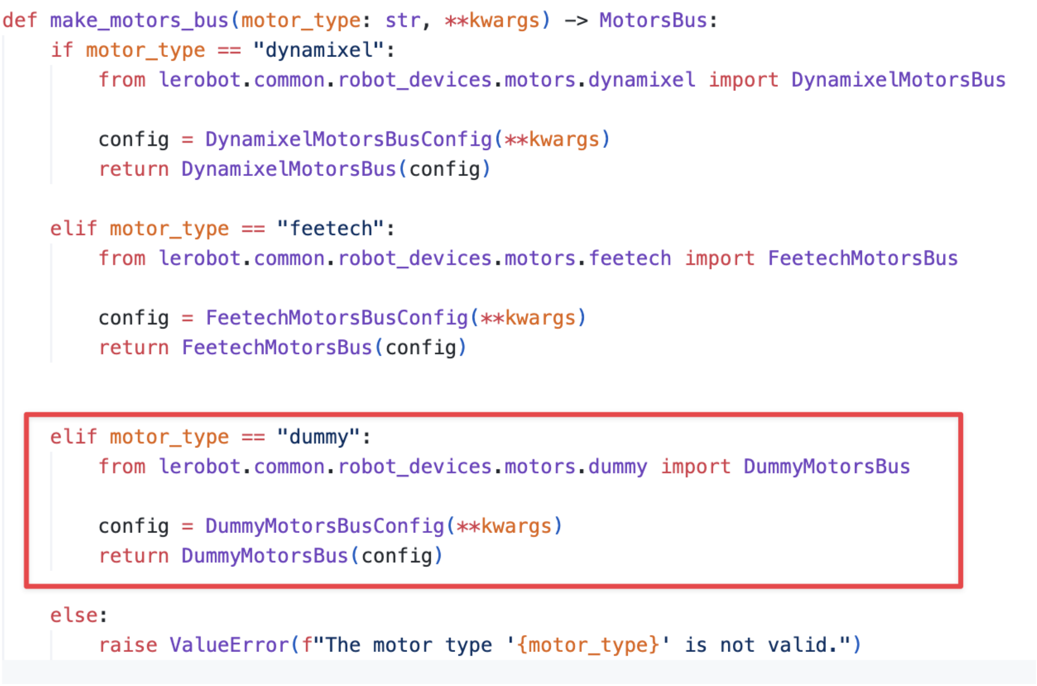
具体实现参考：lerobot/common/robot_devices/motors/

b. 在 `config.py` 中新增 `class DummyMotorsBusConfig`

代码块

```python
@MotorsBusConfig.register_subclass("dummy")
@dataclass
class DummyMotorsBusConfig(MotorsBusConfig):
    port: str
    motors: dict[str, tuple[int, str]]
    mock: bool = False
```

c. 修改 `utils.py` 中的 `def make_motors_buses_from_configs` ，支持从 `DummyMotorsBusConfig` 创建 `DummyMotorsBus`



```
def make_motors_bus(motor_type: str, **kwargs) -> MotorsBus:
    if motor_type == "dynamixel":
        from lerobot.common.robot_devices.motors.dynamixel import DynamixelMotorsBus

        config = DynamixelMotorsBusConfig(**kwargs)
        return DynamixelMotorsBus(config)

    elif motor_type == "feetech":
        from lerobot.common.robot_devices.motors.feetech import FeetechMotorsBus

        config = FeetechMotorsBusConfig(**kwargs)
        return FeetechMotorsBus(config)

    elif motor_type == "dummy":
        from lerobot.common.robot_devices.motors.dummy import DummyMotorsBus

        config = DummyMotorsBusConfig(**kwargs)
        return DummyMotorsBus(config)

    else:
        raise ValueError(f"The motor type '{motor_type}' is not valid.")
```

2. `lerobot/common/robot_devices/robots/`
   a. 创建 `dummy.py`：实现 `class DummyRobot`

代码块

```
class Robot(Protocol):
    robot_type: str
    features: dict

    def connect(self): ...
    def run_calibration(self): ...
    def teleop_step(self, record_data=False): ...
    def capture_observation(self): ...
    def send_action(self, action): ...
    def disconnect(self): ...
```

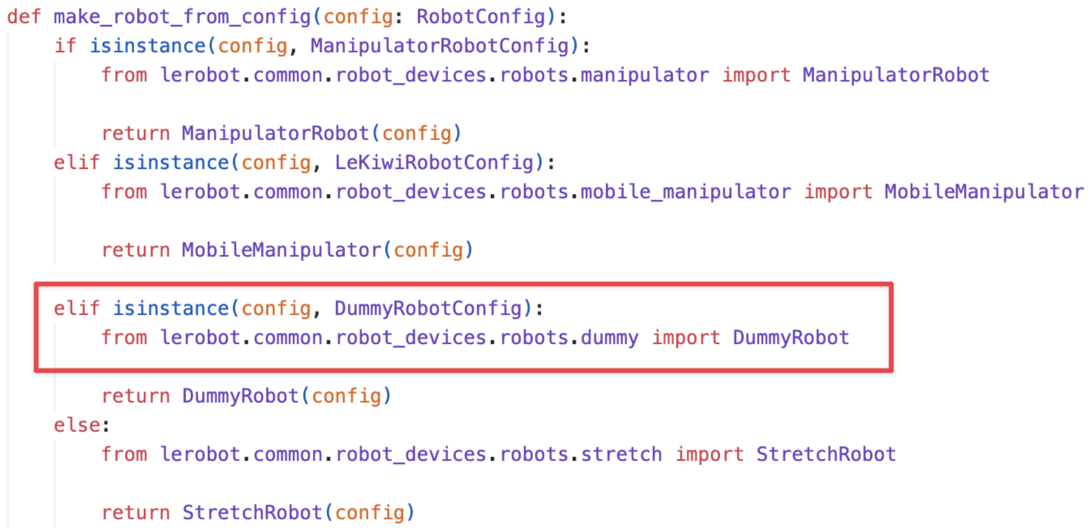
具体参考dummy.py

   b. 在 `config.py` 中新增 `class DummyRobotConfig`，定义 camera 和 motor 类型并配置相关参数

代码块

```
@RobotConfig.register_subclass("dummy")
@dataclass
class DummyRobotConfig(RobotConfig):
    """
    Dummy机械臂配置类
    基于fibre连接的机械臂，通过serial_number连接
    """
    inference_time: bool

    leader_arms: dict[str, MotorsBusConfig] = field(
        default_factory=lambda: {
            "main": DummyMotorsBusConfig(
                port="208C31875253",  # 示例，实际应该替换为真实的序列号
                motors={
                    # name: (index, model)
                    "joint_1": [1, "sts3215"],
                    "joint_2": [2, "sts3215"],
                    "joint_3": [3, "sts3215"],
                    "joint_4": [4, "sts3215"],
                    "joint_5": [5, "sts3215"],
                    "joint_6": [6, "sts3215"],
                    "gripper": [7, "sts3215"],
                },
            ),
        }
    )

    follower_arms: dict[str, MotorsBusConfig] = field(
        default_factory=lambda: {
            "main": DummyMotorsBusConfig(
                port="396636713233",  # 示例，实际应该替换为真实的序列号
                motors={
                    # name: (index, model)
                    "joint_1": [1, "sts3215"],
                    "joint_2": [2, "sts3215"],
                    "joint_3": [3, "sts3215"],
                    "joint_4": [4, "sts3215"],
                    "joint_5": [5, "sts3215"],
                    "joint_6": [6, "sts3215"],
                    "gripper": [7, "sts3215"],
                },
            ),
        }
    )

    cameras: dict[str, CameraConfig] = field(
        default_factory=lambda: {
            "cam_head": NetworkCameraConfig(
                url="http://192.168.237.100:8080/?action=stream",  # 使用IP摄像头URL
                fps=30,
                width=1280,
                height=720,
            ),
            # "cam_head": NetworkCameraConfig(
            #     url="http://192.168.237.157:8080/?action=stream",  # 使用IP摄像头URL
            #     fps=30,
            #     width=1280,
            #     height=720,
            # ),
        }
    )

```

c. 修改 `utils.py` 中的 `def make_robot_from_config`，支持从 `DummyRobotConfig` 创建 `DummyRobot`



```
def make_robot_from_config(config: RobotConfig):
    if isinstance(config, ManipulatorRobotConfig):
        from lerobot.common.robot_devices.robots.manipulator import ManipulatorRobot

        return ManipulatorRobot(config)
    elif isinstance(config, LeKiwiRobotConfig):
        from lerobot.common.robot_devices.robots.mobile_manipulator import MobileManipulator

        return MobileManipulator(config)

    elif isinstance(config, DummyRobotConfig):
        from lerobot.common.robot_devices.robots.dummy import DummyRobot

        return DummyRobot(config)
    else:
        from lerobot.common.robot_devices.robots.stretch import StretchRobot

        return StretchRobot(config)
```

3. `lerobot/common/robot_devices/cameras/`
    a. 创建network.py，支持网络摄像头
    b. 在 config.py中新增 `class NetworkCameraConfig`
    c. 修改 utils.py 中的 `make_cameras_from_configs`